TOPICAL SUBCATEGORY STRUCTURE IN TEXT CLASSIFICATION

RISTO MATTI JUHANI LYRA

submitted for the degree of Doctor of Philosophy

Department of Informatics

School of Engineering and Informatics

University of Sussex

January 2018

DECLARATION

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree.

Brighton, United Kingdom, January 2018

Risto Matti Juhani Lyra

Training is a little like wrestling a gorilla,

you don't stop when you're tired

you stop when the gorilla is tired.

— Fausto Coppi

In memory of my godfather Matti Lamminen.

1958 – 2017

To Ensi and Sara

ABSTRACT

Data sets with rich topical structure are common in many real world text classification tasks. A single data set often contains a wide variety of topics and, in a typical task, documents belonging to each class are dispersed across many of the topics. Often, a complex relationship exists between the topic a document discusses and the class label: positive or negative sentiment is expressed in documents from many different topics, but knowing the topic does not necessarily help in determining the sentiment label. We know from tasks such as Domain Adaptation that sentiment is expressed in different ways under different topics. Topical context can in some cases even reverse the sentiment polarity of words: to be sharp is a good quality for knives but bad for singers. This property can be found in many different document classification tasks.

Standard document classification algorithms do not account for or take advantage of topical diversity; instead, classifiers are usually trained with the tacit assumption that topical diversity does not play a role. This thesis is focused on the interplay between the topical structure of corpora, how the target labels in a classification task distribute over the topics and how the topical structure can be utilised in building ensemble models for text classification. We show empirically that a dataset with rich topical structure can be problematic for single classifiers, and we develop two novel ensemble models to address the issues. We focus on two document classification tasks: document level sentiment analysis of product reviews and hierarchical categorisation of news text. For each task we develop a novel ensemble method that utilises topic models to address the shortcomings of traditional text classification algorithms.

Our contribution is in showing empirically that the class association of document features is topic dependent. We show that using the topical context of documents for building ensembles is beneficial for some tasks, and present two new ensemble models for document classification. We also provide a fresh viewpoint for reasoning about the relationship of class labels, topical categories and document features.

ACKNOWLEDGMENTS

I want to thank my Professor, David Weir, for giving me the opportunity to do a PhD. Thank you for your time and support in iterating through one failed experiment and unworkable idea after another, for having the patience to comment on badly structured drafts and for setting an example on what to aim for. Thank you for maintaining good spirits in the lab, for hosting Christmas and Summer parties and fostering a productive environment.

Thank you Jeremy Reffin, first of all, for convincing David that setting up TAG lab was worthwhile. I doubt I would be submitting anything without your input and (constructive) criticism. I've never once shown you results without you finding a critical flaw in my thinking, an insight in the results I hadn't considered or an altogether new perspective to the thesis. Thank you for the Christmas parties, for hosting Sara and me and for looking after the new recruits, myself included.

Luc Berthouze for supporting me throughout my bachelor's degree and for giving me the opportunity to be a Junior Research Associate, I would not have done a PhD without my involvement in a research project during my Bachelor's degree.

Thank you to Jussi Sarkola and Pentti Halonen for your early involvement in my education. I finally figured out what all that linear algebra can be applied to.

My thesis relies on software that was written by other people and made available at no cost (to me). I owe a great deal to the creators and maintainers of open source libraries especially in the Python community. Thank you to Travis Oliphant for creating NumPy, Travis Oliphant (again), Eric Jones, and Pearu Peterson for SciPy, John Hunter for matplotlib, Fernando Pérez for iPython, Wes McKinney for pandas, Radim Řehůřek and Lev Konstantinovskiy for gensim and Olivier Grisel, Matthieu Blondel, Gilles Louppe, Andreas Müller, Jake VanderPlas, Gael Varoquaux and other core developers of scikit-learn, and of course to the BDFL himself. Thank you also to anyone who has worked on Project Jupyter and to all the people who have contributed code, documentation, bug reports or ideas to the open source projects. Your work in creating open source tools

for scientific computing has allowed me to focus on experimentation and data analysis instead of writing software. Thank you to NumFOCUS and PyData for supporting open source software for scientific computing.

My time at Sussex was significantly ($p = 1^{-15}$, 1-tailed t-test) longer than the three year Bachelor's degree initially intended. I would not have made it through all the set backs, bugs, failed experiments and wrong results without my friends and colleagues by my side. It would have also been lonely and somewhat pointless to celebrate the successful experiments on my own. Thank you to Miro, Sasho, Thomas, Phil, George, Jack, Stana, F. Slorian, Ben, Novi, Viktorija, Emma, Roland, David, David (you know who you are) and Sam – for listening to me, for providing advice when it was needed but not requested, for dragging me out of the office every once in a while and for generally keeping me sane for the past six years. Thank you especially to Sasho and Miro for making me drive across the continent (three times), the holiday was always needed and much appreciated. Thank you to Ben, Miro and F. Slorian for the bike rides. Thank you George for providing me somewhere to stay overnight, for the beer, food and arguments about politics, data structures, networks and everything else. Thank you Thomas and Phil for hosting me and Thomas especially for providing access to his apartment on occasion.

The PyData community specifically in Berlin and generally across the world has been very important for sharing knowledge about experimental methodologies, software best practices but also for the community spirit. I want to thank Katharine Jarmul for her support, encouragement and guidance and for always being the one to put a positive spin on things. Noa, Chris, Adi, Elad, David, Sylvain, Matteo and Adrin for listening to my concerns and for pushing me forwards when I did not have the strength to do so myself. Most of all, thank you for giving me something else to think about when the thesis got to be too much.

The support of my family has been crucial throughout my life, thank you for all the times you bit your lip and did not ask "*how's the PhD going?*", I'll overlook all the times you didn't. Thank you to my grandmother, Maiju, for being interested in what I do and for asking lots of questions, explaining my thesis to you has helped clarify my own thoughts. Thank you to my mother, Tuula, for never giving up on me and for always asking when the next graduation date is. Thank you to my Sister, Anu, for always being

ready to talk about anything. Thank you to my dad, Risto, for all the hard work you've put into raising us. Thank you to my sister, Sanna, for looking after me.

Thank you to my in-laws, Regina and Werner, for all their support and encouragement.

Finally, I want to thank my wife, Sara, for all of the above and below. Thank you for coming along for the ride, for the mental, financial, physical and meta-physical support, for believing in me when I did not and for bringing back down to earth when I got ahead of myself. Thank you for not letting me quit and for sharing the successes as well as the failures. Thank you for your patience and understanding during the three years that turned into four that turned into six, thank you for never quitting on me. Your unconditional, unwavering support for yet another great idea of mine is the reason this document exists.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AdaBoost Adaptive Boosting

LSA Latent Semantic Analysis

pLSA probabilistic Latent Semantic Analysis

LSI Latent Semantic Indexing

IR Information Retrieval

SVD Singular Value Decomposition

LDA Latent Dirichlet Allocation

sLDA supervised Latent Dirichlet Allocation

DiscLDA Discriminative Latent Dirichlet Allocation

MedLDA Maximum entropy discrimination LDA

FLDA Frequency LDA

DFLDA Dependency Frequency LDA

MLTM Multi Label Topic Model

MCMC Markov Chain Monte Carlo

MLP Multilayer Perceptron

PMI Pointwise Mutual Information

TF Term Frequency

TF-IDF Term Frequency Inverse Document Frequency

SVM Support Vector Machine

FSVM Fuzzy Support Vector Machine

MCC Matthew's Correlation Coefficient

OvR One versus Rest

OvO One versus One

i.i.d. independently and identically distributed

ID3 Iterative Dichotomiser 3

CART Classification and Regression Tree

Extra Trees Extremely Randomised Trees

PCA Principal Components Analysis

KNN K Nearest Neighbours

RAKEL Random K Labelsets

AUC Area Under the ROC Curve

ROC Receiver Operating Characteristic (a graph showing false positive rate against the true positive rate)

ML-KNN Multi-label K Nearest Neighbours

HMM Hidden Markov Model

CNN Convolutional Neural Network

MeSH Medical Subject Headings

Part I

INTRODUCTION

1

INTRODUCTION

Document classification is an active field of research with a long history. It has been applied to many problems from early email sorting systems, spam filtering and more recently sentiment classification, relevancy filtering and topic annotation. To perform these tasks automatically an algorithm is given labelled data to establish statistical patterns between document contents and target labels. This mapping usually associates occurrence patterns of words to a class label such as Positive or Negative sentiment, Relevant or Not Relevant in the context of relevancy filtering or possibly a category label such as Sports, Weather, Travel or Economics.

A key factor in creating document classification systems is the relationship between the contents of a document and the class label. Considering the class labels above, we see that there are at least two different types of document classification. The last example with labels such as Travel and Weather is a task that is inherently topical, i.e. assigning a class label depends on the *topical content* or what the document is about. This stands in contrast to sentiment classification, which requires understanding the *propositional content* (Lyons, 1995)[1]: a document expresses positive or negative sentiment towards a target entity through the author proposing negative or positive sentiment rather than a topic the document discusses.

These two types of tasks are sometimes differentiated in the literature as document categorisation for topical content and document classification for propositional content although the terms are often used interchangeably. The distinction, however, is important as it helps us understand the kind of variability a machine learning model is likely

1 In addition to propositional content sentiment classification is considered to require accounting for extra-propositional content, such as author attitude, framing, irony and uncertainty. We review these issues in more detail in Section 2.1.4.

to encounter. We are especially interested in the variability of the statistical patterns between document content and the target labels. Since these patterns are the signal that classification algorithms rely on to decide which label to apply, they need to be consistent across a model's entire life cycle and application domain in order for the model to be useful in practice. However, inconsistencies in the classification signal can arise due to topical diversity.

Data sets with rich topical structures are common in many real world text classification tasks, but many standard document classification algorithms do not take this factor into account. Instead, classifiers are often trained with the assumption that topical diversity does not play a role. However, there is good evidence to suggest the opposite[2] due to the target labels being expressed in different ways under different contexts.

To understand the role topical context plays in shaping how the target label is expressed consider Game Console product reviews and a classifier trained to differentiate positive from negative sentiment. The model will learn that typically negative sentiment is associated with certain words while positive sentiment is associated with others. Those words, however, are not guaranteed to have the same class association in a different context. For instance *restart* could have a negative association for reviews under the Game Consoles category as people complain about having to restart their console, but the same feature can have no class association under the Books category. This has two consequences: first, a classifier trained on Game Console reviews incorrectly interprets the classification signal when applied to Book reviews. Second, a classifier trained on both Game Console and Book reviews would lose a part of the signal that is useful for only a portion of the corpus as the overall corpus wide statistics do not support using *restart* as a feature for the negative class. These issues are aggravated for words with opposing class associations across topical boundaries.

This thesis is focussed on how changes in topical context can impact the performance of standard classification models and how the topical information could be used to improve performance on topically rich datasets. An example of this is Domain Adaptation (Blitzer, Dredze, and Pereira, 2007) where a classifier trained on data from one domain is applied to data from a different domain. The assumption in domain adaptation is that

2 In Domain Adaptation tasks it is well known that the similarity of a source domain to a target domain impacts the difficulty of adapting a classifier from one to the other. For instance Blitzer, Dredze, and Pereira (2007) show that a sentiment classifier trained on book reviews performs worse on kitchen reviews than a classifier trained on electronics reviews.

the source and target domains are known, however, there are many scenarios where a topically diverse corpus exists but the exact topical composition is not known. For instance, many product review websites allow users to submit reviews for a variety of product categories without explicitly defining a category. Similarly data collection methods from online sources often do not allow one to define explicit topical structures. Data collection from Twitter has to happen using a boolean keyword query; in order to collect a large enough sample of documents related to a particular study the queries typically need to be broad and fairly relaxed. The broad, relaxed queries in turn mean that a lot of documents unrelated to the task end up in the data set thus broadening the topical scope of the corpus.

In some cases the topically diverse data sets are cleaned in a pre-filtering stage where documents are classified into Related and Unrelated classes. Documents that are unrelated can be discarded, simplifying later steps in the processing pipeline. This is illustrated in the ACL Shared Task on Aspect-based Sentiment in Social Media Customer Feedback, or GermEval 2017[3]. The task is to extract aspect based sentiment from customer tweets about the service of the German train operator Deutsche Bahn. The task is divided into 4 sub-tasks, the first of which is relevance classification: "*Determine whether a social media post contains feedback about the "Deutsche Bahn" or if the post is off-topic/contains no evaluation*". In other words, discard data that was not intended to be collected in the first place. Note that the algorithm deployed in the pre-filtering stage needs to deal with topical diversity and has to be able to determine the relevance of documents for potentially topically very rich data[4].

Applying standard classification algorithms to these kinds of topically diverse corpora can be challenging due to the class association of words changing across topical contexts. Ensemble models offer a solution for restoring the consistency. Ensemble models were developed to address issues with overfitting: classifiers learning to replicate statistical anomalies in training data that are not useful overall for the task. Traditionally, ensembles are trained by taking repeated random samples from the training data and building multiple models, one per sample. The data samples are taken at random to guarantee

3 https://sites.google.com/view/germeval2017-absa/

4 A motivating factor for the thesis was a project that involved building a pre-filtering system in a media monitoring setting (Lyra et al., 2013)

that the statistical properties of the samples reflect those of the original data set and due to a lack of prior knowledge about the data being sampled.

Given that document classification deals with natural language we do have prior knowledge about the data, more specifically, we know that natural language is topical. The topic of a discussion or a document constrains the vocabulary to revolve around the topic itself; an article about the World Cup final is likely to refer to goals, footballs, player groupings and team tactics and less likely to refer to black holes, anti gravity or German politics.

The aim of this thesis is to explore whether the topicality of natural language can be used to guide building ensemble classifiers; in Chapters 3 and 4 we develop novel ensemble methods that rely on a topical decomposition of a corpus. In Chapter 3 we focus on a sentiment analysis task and build an ensemble from linear classifiers together with topic modelling[5]. In general, the ensemble consists of linear classifiers that are biased towards specific topics during training by modifying the objective function of the classifier. In Chapter 4 we focus on a hierarchical multi-label classification task and develop a new learning framework for topical ensemble models. Our framework combines topic modelling with an efficient weight computation and significantly outperforms several comparison models.

The contribution of this thesis is as follows:

1. We show that the class association of word types is topic dependent, and that accounting for topical context is beneficial for some tasks.

2. We develop a novel ensemble method based on topic modelling and linear classifiers for document classification.

3. We show that using an ensemble method for tasks that depend on propositional content is beneficial.

4. The empirical findings further show that tasks that require understanding propositional content, such as sentiment analysis, do not necessarily benefit from the topical information although using an ensemble model improves performance. This finding suggests an alternative explanation to the improvement shown by Xiang and Zhou, 2014.

5 We use Support Vector Machines and Latent Dirichlet Allocation as the building blocks for the ensemble.

5. We present a novel ensemble method for hierarchical multi-label classification and show that it significantly improves performance over baseline models in a real world task.

6. We present a "vocabulary agreement" metric for measuring how much the classification signal diverges in a binary task between two different application domains.

The rest of this thesis is structured as follows: Chapter 2 reviews the relevant literature, and presents the algorithms used in this thesis. Chapter 3 addresses the problem of sentiment classification and develops a topical ensemble model for that scenario. Chapter 4 deals with hierarchical multi-label document classification and presents a novel ensemble classifier based on a topical division of the data. Chapter 5 concludes the thesis and discusses future research directions.

2

RELATED WORK

This thesis focuses on document classification of datasets with a broad topical range. We investigate how the topical information can be utilised to improve the performance of a classification algorithm. The two tasks we focus on are sentiment classification and hierarchical multi-label classification. In sentiment classification we focus on document level binary sentiment in user generated product reviews, and in multi-label classification we focus on topical categorisation of newswire text. These tasks have normally been performed using algorithms that are insensitive to topical context, but recent research has investigated how topical context could be used to benefit classification methods. We present methods that add to the existing literature by using topical information and embedding topically biased classifiers in classifier ensembles.

This Chapter provides context to our work by presenting previous research in document classification, ensemble models and topic models, and is structured as follows. Sections 2.1 and 2.2 give background information for understanding document classification and give an overview of sentiment classification. Section 2.4 describes both foundational research in model ensembles and more recent research in building model ensembles for topically rich document classification scenarios. Section 2.5 describes topic models and how they have been used previously to address the tasks that are the focus of this thesis. Section 2.6 covers the kinds of modifications made to standard algorithms to handle multi-label problems and provides context for the kinds of application scenarios where multi-label classification is needed.

In the discussion until now we have used the term "document classification" as a general concept. Since this thesis explores utilising topical information to improve classification algorithms it is important to make a note on terminology: the terms *classifi-*

cation and *categorisation* are used somewhat interchangeably by researchers. Most use *categorisation* when the task is to assign documents into topical categories, tasks where the categories correspond to our intuitive notion of a topic such as Politics, Sports or Weather. When the task is to assign documents to non-topical groups such as Positive or Negative sentiment the task is normally referred to as *classification*. Classification, however, is sometimes also used to refer to the former and as such is a term that encompasses all tasks where a label or collection of labels is assigned to documents. We will use *class* or *target label* to refer to the target labels of a classifier, whether the target labels are topical or non-topical in nature. It will be clear from context which kind of target label is being discussed. It is important to note that in some scenarios the category or topic of a document is aligned with the class of the document and in others it is not. In sentiment analysis, for instance, the class is different from the category or topic.

Finally, *category* and *topic* are often also used interchangeably. We will use *category* to refer to a human understandable semantic grouping such as Politics or Weather and *topic* to refer to the output of a probabilistic topic model.

SECTION 2.1
Sentiment Classification

Manufacturers, retailers and service providers are interested in what customers think of their product or service. Managing the brand identity is important in a competitive marketplace and customer loyalty can be increased by promptly addressing questions or grievances. The frustrations and grievances as well as positive feelings are often expressed on social media or on review websites, making those platforms useful resources for detecting positive or negative sentiment towards a product or company. Sentiment Classification (Turney, 2002; Pang, Lee, and Vaithyanathan, 2002; Pang and Lee, 2004) is a general term for a number of different classification tasks. At its simplest Sentiment Classification is the task of automatically assigning a positive or negative sentiment label to a document such as a product review or a tweet. Sentiment Classification can also be a multi-class task where the class labels can be positive, negative and neutral (Poursepanj, Weissbock, and Inkpen, 2013; Chalothorn and Ellman, 2013; Becker et al., 2013) or an even finer grained sentiment polarity task with five classes, 13 classes (Chaovalit and Zhou, 2005) or possibly a continuous rating scale for positive or negative sentiment.

Sentiment classification is diverse also in the types of data the methods are applied to. Sentiment classification can be done on the document level, as is the case in this thesis, or on smaller units such as sentence level or word level. Sentiment analysis can also be oriented towards specific aspects of a product or a service (see Section 2.1.2). This Section gives an overview of the types of sentiment classification tasks that researchers have looked at in the past and the methods proposed to solve those tasks.

Seminal work in Sentiment Classification by Turney (2002) used heuristic methods to extract sentiment containing phrases and Information Retrieval (IR) methods to associate sentiment to the extracted phrases. They first extract subjective language (See Subsection 2.1.4) and then score the extracted phrases (bi-grams) according to their semantic orientation. The semantic orientation is calculated as the difference between the Pointwise Mutual Information (PMI) score of a phrase with a highly positive word (the word "excellent") and the PMI score with a highly negative word ("poor"). Finally a document is classified as positive ("recommend" in their use case) if the average semantic orientation score of phrases extracted from the document is positive. The work is highly relevant for this thesis as one of the key challenges identified by Turney (2002) is the variability in semantic orientation of sentiment bearing phrases across domains. This variability is a key reason for them to use adjective/noun bi-grams instead of single adjectives as the document features; they specifically note the importance of the contextual information noting that: "... the adjective "unpredictable" may have a negative orientation in an automotive review, in a phrase such as "unpredictable steering", but it could have a positive orientation in a movie review, in a phrase such as "unpredictable plot"". This exact same problem is a motivation for the work in this thesis. We also use a scoring technique that is similar to theirs for finding words that have a high variability in their semantic orientation across different domains (see Section 3.2.1). Our method however is not limited to just positive or negative sentiment but generalises to any binary classification task.

Pang, Lee, and Vaithyanathan (2002) extended the work of Turney (2002) and evaluated standard machine learning models – Naïve Bayes, Maximum Entropy and Support Vector Machines – on movie review data. They evaluate a host of different feature extraction methods and show that machine learning models have competitive performance against human counterparts on a sentiment analysis task on movie reviews. They also

provide a pertinent discussion about how people tend to express opinion noting that simple bag-of-words classifiers are unlikely to excel at the task. Their error analysis shows that the machine learning models commonly make mistakes on documents where a negation of a previous negative sentiment is used as a rhetorical device. Having a better contextual understanding on how sentiment is expressed might help. Various approaches for contextualising how sentiment is expressed have later been proposed both in the field of computer science (Socher et al., 2013) and in linguistics Polanyi and Zaenen (2006) and Benamara, Chardon, Mathieu, et al. (2011). These approaches and more are reviewed in the following 4 Subsections.

2.1.1 *Sentiment Mining and Summarisation*

As product review websites, personal blogs and user forums have gained in popularity the need for mining information from those online sources has grown. A number of systems have been developed to aid aggregating user opinions from online sources. These systems are often motivated by a need to have a complete picture of user opinions. Both Morinaga et al. (2002) and Dave, Lawrence, and Pennock (2003) developed methods for addressing this need and designed systems for searching and displaying the aggregated opinions through a query interface.

Morinaga et al. (2002) presented a sentiment extraction tool that collected opinions about specific products from free text on web pages given specific product names as query terms. Their system extracts opinion expressions based on a pre-defined dictionary and ranks the extracted statements based on how likely they are to contain an opinion. Stochastic complexity (Rissanen, 1996) is used to automatically construct classification rules from statements that both contain an opinion and occur in the vicinity of a product mention. The rules are essentially word lists that contain typical opinion words for certain products. The words alone do not provide enough context for the user of the system to evaluate typical opinions about the target product so a word co-occurrence analysis is used to extract larger contexts for the opinions. Finally, full sentences that contain opinions are ranked based on the information extracted earlier. These sentences are used to contextualise the opinions users typically hold of a certain product. An informal analysis shows that the system is able to extract opinions from free text and provide

some context to product oriented sentiment analysis. However as no quantitative evaluation or comparisons with other systems or baselines is done it is difficult to say how effective the system is.

Dave, Lawrence, and Pennock (2003) developed a tool for aggregating and summarising user generated product reviews from online sources. They first train a Naïve Bayes classifier on user rated reviews from C|net or product reviews from Amazon. They tested a number of feature selection and smoothing techniques and compared the Naïve Bayes classifier to a method that uses averages of feature scores with the Naïve Bayes classifier having better performance. Dave, Lawrence, and Pennock (2003) highlight a few issues that are of interest in the context of this thesis. They note the large number of contexts in which a product is mentioned and specifically note that a specialised genre classifier is likely needed to assign phrases into coherent topical categories. Secondly, as the methods they use are statistical the document features that carry high positive or negative scores can be unexpected features from a linguistic standpoint. For instance, in their study "headphones" was a negatively weighted feature due to the word being used much more in negative reviews. This is an interesting observation as it highlights how some product features tend to be ones the are cited more in negative contexts while other are cited more in positive contexts. The product features themselves are not as such positive or negative, but they can serve as a useful signal of sentiment in statistical models. Although the presented system works well on product reviews the performance degrades when applied to general web content which may or may not contain opinionated text.

Yi et al. (2003) extended the work of Dave, Lawrence, and Pennock (2003) by focussing on extracting sentiment about a given topic[1] and on improving the performance on general web text. They also use linguistic patterns to extract opinion features, but introduce two novel statistical feature selection methods to select and assign polarity scores to the extracted features. The first feature selection method uses multinomial language models – one for general language and one for topic specific language – to compute topic specialised feature scores. The second method uses a statistical likelihood test introduced by Dunning (1993). The final sentiment classification is done using hand crafted heuris-

1 Yi et al. (2003) use the words "topic" and "subject" somewhat interchangeably, but crucially they are not referring to a topic in the strict sense of topic modelling but rather in the more common "a topic of discussion" sense.

tic methods. On product reviews the system has comparable performance with that of Dave, Lawrence, and Pennock (2003), but far outperforms ReviewSeer on general web content.

Hu and Liu (2004) demonstrate a system for summarising user written product reviews that relies on identifying opinionated sentences and the polarity of those sentences before finally producing summaries of the aggregated reviews. The identification of important opinion words is different to Dave, Lawrence, and Pennock (2003) in that specifically adjectives are extracted as opinion words. The polarity of the extracted adjectives is determined from WordNet (Miller et al., 1990) synsets using a set of seed adjectives with known polarity as the basis and expanding those known polarities through the WordNet synonym / antonym adjective graphs. One major issue with the adjective seeding process of Hu and Liu (2004) is that the they come up with the positive and negative adjectives, 30 in total, themselves. This both restricts the applicability of their method to sentiment analysis and does not account for the variability in language use pointed out in previous research (Dave, Lawrence, and Pennock, 2003; Pang, Lee, and Vaithyanathan, 2002; Turney, 2002).

2.1.2 *Fine-Grained Sentiment Analysis and Encoding Lexical Polarity*

An important aspect of sentiment analysis pipelines is the accurate identification of polar expressions and the subsequent encoding of their polarity. A number of studies have taken the approach of Hu and Liu (2004) and used WordNet as a lexical resource to determine the sentiment polarity of lexical items.

Kamps et al. (2004) suggested using the minimum path length of a target word to a known polar adjective ("good" / "bad") as a measure for the target word's semantic polarity and Ding, Liu, and Yu (2008) propose including context dependent opinion words in an opinion lexicon. They determine the contextual polarity of words using linguistic features combined with WordNet (Miller et al., 1990) synonym / antonym pairs. For instance, if the word "long" is found to be positive in the context of "battery life" then long's antonym "short" is assigned negative polarity for that context regardless of "short" having been observed in that context or not. Both Hu and Liu (2004) and

Ding, Liu, and Yu (2008) use their respective methods to extract fine grained sentiment information about product features.

Jin, Ho, and Srihari (2009) presented an integrated machine learning model that learned a Hidden Markov Model (HMM) over a sequence of word part-of-speech tag pairs. To make learning the model parameters computationally feasible the authors restrict the Markov chain to be first-order. They address feature sparsity bu using the polarity propagation methods presented by Hu and Liu (2004). Overall the results show that although statistical machine learning methods are computationally expensive they can also have competitive performance if tuned correctly.

The motivation for extracting sentiment about specific aspects of a product comes from a realisation (Hu and Liu, 2004) that while a review may overall be positive it can contain information about some negative aspects of the product being reviewed. For a manufacturer knowing about these negative aspects is important. The task has later been named Aspect Based Sentiment Analysis (Pontiki et al., 2014). The approaches to Aspect Based Sentiment Analysis have multiplied and range from supervised and unsupervised machine learning approaches (Gupta and Ekbal, 2014; García Pablos, Cuadros, and Rigau, 2014) to extracting syntactic and lexical features (Negi and Buitelaar, 2014; Hangya et al., 2014) to creating distributed word representations (Blinov and Kotelnikov, 2014).

While aspect based sentiment analysis has been established as a discrete sub-task it is not the only fine grained sentiment analysis tasks that has received attention. Socher et al. (2013) introduced a sentiment treebank that builds on the work of Pang and Lee (2005). The data set contains over 10000 sentences with their entire parse tree annotated with fine grained sentiment information. The aim is to facilitate research in compositionality, that is, how the sentiment expressed in a sentence is composed out of smaller units and what impact for instance negation has. Socher et al. (2013) present a neural network model that is trained on the sentiment treebank and show that performance at a binary sentence level classification task increases by $\sim 7\%$ when accounting for the compositionality of sentiment over a sentence.

The advent of neural network approaches to sentiment analysis has shifted the focus away from explicitly extracting and scoring sentiment polarity and towards encoding sentiment information into distributed word representations. These approaches often

do not need complex linguistic feature extraction processes but instead use pre-trained word embeddings (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013; Pennington, Socher, and Manning, 2014), or a task that allows sentiment information to be encoded in the word embeddings during training. Kim (2014) showed that using pre-trained word-embeddings together with a Convolutional Neural Network (CNN) architecture yields competitive performance with the state-of-art and that fine-tuning the word embeddings during training allows further performance gains to be made. The fine-tuning is done simply by allowing the pre-trained word embeddings to be changed based on the task specific loss function and training data.

Ye, Li, and Baldwin (2018) focus on encoding external sentiment information into word embeddings by jointly learning a sentence level sentiment classifier and a separate model that predicts the sentiment distribution of words in the current sentence. Contrary to the fine-tuning method of Kim (2014) this method encodes sentiment information that is not directly dependent on the primary prediction task (sentence level sentiment classification). The approach is interesting as it provides a way of integrating any number of different external information sources. For instance, the method could be used to encode topical information in addition to sentiment information. Their method beats multiple baselines both on document level binary sentiment classification tasks as well as fine-grained sentence level sentiment classification.

2.1.3 *Domain Aware Sentiment Classification and Out-of-domain Data*

Supervised learning methods often require lots of labelled training data. This is often a problematic limitation for sentiment classification systems as labelled training data for many domains is not available or is expensive to acquire. Several studies have focussed on the specific problem of cross domain sentiment classification Bollegala, Weir, Carroll, and Ishizuka (2010), Zhou et al. (2015), Zhang, Hu, et al. (2015), and Bollegala, Weir, and Carroll (2011). In this task training data exists or can be labelled for a source domain but not for a target domain. Since sentiment is expressed in different ways between the domains this presents at least two challenges: features observed in one domain may be absent from another and the polarity of features under one domain can be different in the other.

Aue and Gamon (2005) showed that a standard Support Vector Machine (SVM) classifier suffered a very substantial drop in accuracy when applied to a domain the classifier was not trained on. They consequently conducted a number of experiments comparing different ways of combining labelled and unlabelled data from the source and target domains and an ensemble model to overcome the domain specificity. Their results indicate that using data, either labelled or unlabelled, from the target domain is beneficial. In all experiments the best performing model was a modified Naïve Bayes classifier (Nigam et al., 2000) that utilises information from unlabelled instances sampled from the target domain.

Read (2005) showed that sentiment classifiers are both training domain and time dependent, i.e. classifiers are biased not only by the training domain but also by the temporal sentiment trends present in the training corpus. They proposed using an external source of information to train a sentiment classifier that should allow the classifier to be less dependent on the domain of the training data. They extracted Usenet discussions that contain emoticons and used a context window around positive / negative emoticons to train a sentiment classifier. The trained classifier was then applied to test data from a different domain for instance movie reviews. Their results show that although the Usenet emoticon based classifier is effective in classifying in-domain data it performs only slightly better than a random classifier on the out-of-domain target data. The poor performance is attributed to noise in the Usenet data and poor coverage of tokens in the test data.

Both Wu, Tan, et al. (2009) and Ponomareva and Thelwall (2013) used graph based methods to approach the problem of cross domain sentiment analysis. Wu, Tan, et al. (2009) used an approach that is conceptually similar to a K Nearest Neighbours (KNN) classifier (Cover and Hart, 1967). Their method assumes that a training data set and a test data set come from different but related domains, and that the sentiment labels of both training and test documents are similar to those of the documents' neighbours. In order to build the document graph, similarities between the documents must be computed. This is done using cosine similarity on Term Frequency Inverse Document Frequency (TF-IDF) document vectors (see Section 2.2.1.1). The assumptions made by Wu, Tan, et al. (2009) are somewhat counter to the hypothesis of this thesis. For their method to work well the sentiment labels of two documents would need to have a near linear relationship

with the overall similarity of the documents. However, defining document similarity is a tricky task (see Section 2.5.3.3) and there is no reason to assume that two documents that, for instance, discuss the Brexit vote share their sentiment polarity about the topic. Indeed, Ponomareva and Thelwall (2013) specifically point out that in order for the graph methods to be meaningful "... [the] vector representation of the data must contain sentiment markers rather than topic words". Nevertheless the results of Wu, Tan, et al. (2009) indicate that their method is at least better than the baselines it was compared to. Ponomareva and Thelwall (2013) further explore using graph methods for cross domain sentiment analysis. They use a slightly modified label propagation technique (Zhu and Ghahramani, 2002) and show competitive results on a small sentiment analysis dataset.

More recently Wu, Huang, and Yuan (2017) address the problem of domain specific sentiment classification by fusing information from multiple sources. Their approach is motivated by the understanding that sentiment is expressed in different ways in different domains, and that a machine learning model for sentiment classification in one domain is not necessarily suitable for data from a different domain. Their approach uses labelled data in both the target and source domains as well as external sentiment lexicon features. The sentiment of words contained in an external sentiment lexicon can be propagated to domain specific sentiment words not found in the lexicon by looking at the frequency at which the words co-occur in certain contexts. Their results beat multiple baselines on an Amazon product reviews data set.

Out-of-domain data in the context of sentiment classification is often understood to be data from a different online source or topical category. For instance, Read (2005) used data from Usenet and movie reviews, whereas Ponomareva and Thelwall (2013) and Blitzer, Dredze, and Pereira (2007) used different product categories. An alternative take on different domains is exemplified in the NTCIR-6, NTCIR-7 and NTCIR-8 cross-domain classification tasks (Seki, Evans, Ku, Chen, et al., 2007; Seki, Evans, Ku, Sun, et al., 2008; Seki, Ku, et al., 2010). In these tasks the domains reach across language boundaries. For instance, the multilingual opinion question answering sub-task in NTCIR-8 involves questions in English and answers in traditional Chinese, simplified Chinese and Japanese. The sub-tasks also include finding opinion holders and opinion targets. These tasks show how varied sentiment analysis as an application field can be. The best performing approaches to NTCIR-8 were similar to the ones already presented

here: feature extraction and filtering based on linguistic features and polarity scoring based on external lexical resources (Balahur et al., 2010).

2.1.4 *Subjective Language*

Sentiment analysis concerns itself with finding and correctly classifying language that expresses sentiment. As such sentiment analysis is related to recognising subjective language. Wiebe et al. (2004) argue that sentiment analysis methods can benefit from specifically accounting for subjective language and should, in order to function properly, do so. Polanyi and Zaenen (2006) further argue that only focussing on the polarity of individual terms often leads to an incorrect interpretation of the meaning of the text. This Subsection gives a brief overview of the research conducted in recognising subjective language in so far as it related to this thesis. For a thorough review of subjectivity and sentiment we refer the reader to Benamara, Taboada, and Mathieu (2017).

Wiebe et al. (2004) argue that many subjective expressions also have objective usages and that recognising subjective language is therefore context dependent. The authors focus on extracting subjective language based on three different clues: words that occur only once in a corpus, collocations that are indicative of subjectivity and distributional similarity (Lin, 1998). All three methods improve the precision of identifying subjective language over a baseline model.

Riloff and Wiebe (2003) developed a statistical framework for learning linguistic patterns of subjective language. In their framework high precision classifiers that rely on pre-existing word lists are used to separate subjective from objective texts; these sets of texts are then used to learn new extraction patterns from subjective language by computing the conditional probability of a subjective expression given the subjective and objective texts. The subjectivity patterns extracted by Riloff and Wiebe (2003) were later used by Wilson, Wiebe, and Hoffmann (2005) in a hierarchical classification model for sentiment analysis. The authors first identify subjective language phrases and then disambiguate those phrases into positive and negative sentiment.

Language is not necessarily subjective only when explicit subjective phrases are used. Subjective language can be much more subtle as demonstrated by Polanyi and Zaenen (2006). They argue that the ways in which "...lexical items reflect attitudes is more

complex than simply counting the valences of terms". Their central argument is that the valence[2] of lexical items can be strengthened or weakened by cultural context the presence of other lexical items and the discourse structure of the text. The work provides crucial context for why sentiment analysis is difficult, and more specifically, why sentiment analysis on corpora with complex topical structures is difficult. Among the aspects that change the valence of lexical items are: modals[3], connectors that modify or negate the meaning of an utterance ("Although Boris is brilliant at math, he is a horrible teacher."), genre constraints and cultural context. Genre constraints is of particular interest here as Polanyi and Zaenen (2006) point out that movie reviews[4] tend to have two types of information: information about the movie plot, locations and characters and information about the production of the movie. For sentiment analysis it is the latter information that is of relevance.

Benamara, Chardon, Yannick, et al. (2012) looked more specifically at the issue of negation and modality. They presented native speakers with sentences that contain (do not contain) negation or modal expressions and asked the native speakers to score opinionated sentences or if the sentence expresses an established opinion. Their empirical work shows that negation affects both the polarity and strength of opinion expressed and that modality impacts the strength of opinion expressed.

Recasens, Danescu-Niculescu-Mizil, and Jurafsky (2013) extended the discussion around subjective language to the more general biased language which includes linguistic traits such as epistemological and framing bias. Epistemological bias is exhibited in statements that are commonly held to be true (false) and are presupposed in word choices. For instance, "he believed sentiment analysis to be a complex problem" versus "he realised sentiment analysis was a complex problem". The use of "realised" in the latter sentence implies that he came to understand a universal fact as opposed to just holding a personal belief. Framing bias refers to the usage of terms that reveal the writer's stance on a particular issue; a good example of this are the terms "pro-life" and "anti-abortion". Although the terms both refer to the same political stance the former contextualises that stance positively while the latter does the opposite. Their method works in three steps:

2 Valence is the negative or positive attitude that a lexical item communicates.

3 The example given in the paper is the difference in sentiment between "Mary is a terrible person. She is mean to her dogs." and "If Mary were a terrible person, she would be mean to her dogs." The latter sentence does not suggest that Mary is either a terrible person or mean to her dogs.

4 At the time of publication the movie review data set of Pang, Lee, and Vaithyanathan (2002) was the de facto standard for testing sentiment analysis methods.

finding biased phrases, identifying the biased terms in the phrase and finally correcting the biased terms. They apply a logistic regression model to the task and show that their model is very close to the human baseline performance.

Benamara, Chardon, Mathieu, et al. (2011) used methods from discourse analysis to perform subjectivity analysis. They extracted discourse fragments which were then used to contextualise the subjectivity / objectivity of statements within the fragments. They also provide a four-way breakdown of the ways in which sentiment can be expressed within the discourse segments: explicit positive or negative sentiment statements, positive or negative statements implied in an objective statement, non-subjective statements and subjective statements that do not express sentiment.

Recently Abdul-Mageed (2018) compared deep learning models with carefully feature engineered SVMs on Arabic language Tweets. They show that using social context and structural features such as hashtags, URLs, quotations etc. for SVMs significantly improves performance over a baseline model and that recurrent neural networks further improves the prediction accuracy for subjective language.

SECTION 2.2
Algorithms for Document Classification

In this Section we formally define the problem of statistical document classification and outline common method for representing documents in machine learning tasks. Our focus is on methods that address the application areas dealt with in this thesis and methods that share similarities to those presented within it. For a complete historical perspective on feature selection we refer the reader to existing literature by Aggarwal and Zhai (2012), Shalev-Shwartz and Ben-David (2014), Manning, Raghavan, and Schütze (2008), and Forman (2003).

2.2.1 *Statistical Document Classification*

Statistical document classification is the problem of determining which class label(s) $y \in Y$ to assign to instances $X = \{\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n\}$. The set Y is a discrete set of class labels of size $|Y|$, and the documents X are normally represented as a vector of document features $\mathbf{x} \in \mathbb{R}^N$ or $\mathbf{x} \in \{0, 1\}^N$. The dimensions N typically encode words in a document

but can also represent other information such as latent topics inferred using a topic model[5].

To perform automatic document classification a statistical model is trained from a labelled data set D_l; the set D_l consists of pairs documents with their correct labels $D_l = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_n, y_n))$. Normally only one label from Y is assigned to each document. Multi-label classification is an exception; in multi-label classification any number of labels can be assigned to a document.

Before a machine learning algorithm can learn a mapping from text to labels, the raw text in documents has to be transformed into some document representation. The general description of document classification above leaves out the details of what the feature vectors $\mathbf{x} \in X$ are composed of and as a result what kind of a signal the document representation provides. The document representation needs to carry a consistent signal of the target classes in order for the algorithms to learn something useful. This signal is determined by the preprocessing steps applied to the documents to extract document features. Some preprocessing methods are designed to separate signal from noise while others aim to highlight unique document features. The following three Subsections briefly cover the areas of feature extraction, feature values and feature selection.

2.2.1.1 *Feature Extraction*

The most common method for transforming raw text into feature vectors is the bag-of-words representation. This representation encodes the vocabulary of a dataset as word index locations and each document as a vector whose length is equal to the size of the vocabulary; words (features) present in a document are marked with a positive integer and everything else is 0. Most entries in a document's feature vector will be zeros and usually only a handful will be non-zero.

The bag-of-words representation encodes documents in a purely symbolic form: words that are meaningful to humans are encoded as unique index positions that carry meaning only through their relation to the target classes. That relation is established using Term Frequency (TF) feature values that encode the frequency of words in each docu-

5 Modern neural network based approaches often do not have an explicit document representation, instead the algorithms consume the input one document feature at a time generating an implicit document representation (Howard and Ruder, 2018).

ment[6]. TF values are sometimes normalised by the length of the document to account for differences in document length.

Feature values can also be transformed to carry information about words that are noteworthy in a document in relation to the rest of the dataset. A common method for achieving this is the TF-IDF transformation (Equation 1). TF-IDF highlights words that are prominent in a specific document (Manning and Schütze, 1999; Manning, Raghavan, and Schütze, 2008).

$$\mathrm{tfidf}(w, \mathrm{d}; D) = \mathrm{tf}(w, \mathrm{d}) \log \frac{\mathrm{N}}{|\{\mathrm{d} \in D : \mathrm{t} \in \mathrm{d}\}|} \tag{1}$$

TF-IDF is not a method for finding features that distinguish the target labels from each other. Instead, it creates a document representation where statistically salient words in a document are highlighted. TF-IDF is therefore an unsupervised method and does not require any labelled training data for learning the feature weights. However, training a supervised machine learning model using TF-IDF feature weights does require labelled training data.

Depending on the task some features are more informative than others. The methods described above are not designed to optimise the document representation for a classification task or to maximise the informativeness of features w.r.t. target labels. To make the signal cleaner feature selection methods have been developed. Those method are described in the next Subsection.

2.2.1.2 *Feature Selection and Dimensionality Reduction*

To help algorithms distinguish between target classes, researchers have developed methods for selecting a good subset of features (Forman, 2003; Aggarwal and Zhai, 2012; Forman, 2003; Guyon and Elisseeff, 2003). The methods require labelled training data to determine which features, or combinations of features, best represent a target class. Guyon and Elisseeff (2003) divide feature selection methods into three groups: filtering methods, wrapper methods and embedded methods. We give a brief overview of methods in these three groups.

6 A simplification of TF feature encoding is to use Boolean feature value that only encode the presence of features, not their count in documents.

Filtering methods rely on a statistical measure of association. The association is computed between a feature w and a target class c for all features independently or in some cases for groups of features. Typical measures include the χ^2-statistic, Mutual Information, Information Gain and Gini Index. The metrics rank features according to how well the features differentiate the target labels.

Given a ranking of all features based on their distinctiveness the top N features, or a certain percentage of features, are then selected as the final representation. In some cases combinatorial groups are also tested, but these methods tend to be computationally expensive and thus intractable for many real world problems (Recursive Feature Elimination by Guyon, Weston, et al. (2002)).

Wrapper methods work together with a predictor to select features that maximise the predictor's performance on some test data set. The predictor is used as a black box to evaluate the performance of feature subsets. Since the underlying predictor is treated as a black box, wrapper methods can be used with any predictor, but the selected features are specific to the predictor and do not necessarily generalise to other prediction algorithms.

Embedded feature selection methods are particular to specific learning algorithms. For instance, decision trees (see Section 2.3.1) perform feature selection by splitting the training data based on entropy or Gini Index. Similarly, Support Vector Machines with L1-regularisation implicitly perform feature selection by setting some model coefficients to 0.

The feature selection methods can be seen as a way of reducing the dimensionality of the feature space. Dimensionality reduction can also be done using methods that capture contextual information about word co-occurrence. Word co-occurrences reflect how writers use language to talk about certain topics and allow researchers to model topical information in an unobservable latent space. Methods such as Latent Semantic Indexing (LSI)[7] (Deerwester et al., 1990) and Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan, 2003) create dense feature vectors of a few hundred dimensions for each document and have been used as dimensionality reduction in document classification tasks. We will cover both LSI and LDA in detail later in Section 2.5.

7 Latent Semantic Indexing is also known as Latent Semantic Analysis (LSA), but LSI is the preferred term in information retrieval and related communities.

The main issue with these latent methods is that the final document representation is not intended to be useful for the classification task itself, but to reduce the dimensionality of the feature space. The utility of the method(s) is therefore dependent on the application scenario. This thesis is focussed on exploring how topical information can be used to benefit different kinds of document classification tasks.

SECTION 2.3
Classification Models

In this Section we introduce the models commonly used in document classification. We focus on classification algorithms that are used either as comparison models or as part of the topical ensembles in Chapters 3 and 4. Model ensembles are covered in Section 2.4.

2.3.1 *Decision Trees*

Decision Trees are a general purpose method for classification and regression that rely on rules organised in a tree structure; each node is a decision point based on one feature in the dataset. Documents that have the feature[8] are passed down one side of the tree whereas those without the feature are passed down the other side. Following a particular path down the tree filters the dataset based on a combination of features. Leaf nodes at the bottom of the tree apply a label to a document.

A learning algorithm (Iterative Dichotomiser 3 (ID3), C4.5, Classification and Regression Tree (CART)) iteratively builds the decision tree data structure by splitting the training data into two parts using an information theoretic metric. The metric computes how informative document features are of the target classes. The decision rule that maximally separates the remaining training data at each node is selected as the decision rule at that point.

Different learning algorithms have different ways of dividing the input feature space. ID3 (Quinlan, 1986) and later C4.5 (Quinlan, 1993) build decision trees by splitting the training data based on the entropy of class labels on one side of a potential split. At each

8 Generally the decision points are based on thresholds with feature values less than the threshold passed down one side.

node in the tree the algorithm cycles through all previously unused features and selects the one which minimises entropy, defined as

$$\phi_H(D_m) = -\sum_{y \in Y_m} p(y) \log_2 p(y) \tag{2}$$

where D_m is defined as the dataset at node m and Y_m is the set of class labels in D_m. $p(y)$ is the marginal likelihood of class labels $y \in Y_m$.

C4.5 extends the ID3 algorithm by adding support for continuous variables through thresholding and by pruning sub-trees whose absence does not increase the error rate.

CART trees (Breiman et al., 1984) are similar to ID3 and C4.5 but instead of minimising the entropy CART trees minimise the Gini index (Equation 3). CART trees also support both categorical and continuous target variables unlike C4.5.

$$\phi_{gini}(S_m) = \sum_{y \in Y_m} p(y)(1 - p(y)) \tag{3}$$

Decision trees create models that are easy to interpret and visualise. This can be beneficial in some application scenarios; in classification tasks the learned decision trees can also be used to aid in feature selection for other classification algorithms (Sugumaran, Muralidharan, and Ramachandran, 2007; Questier et al., 2005). To the best of our knowledge this feature selection method has not been applied to document classification.

2.3.2 *Logistic Regression*

The logistic regression classifier is a simple and effective method for binary classification. We follow the definition of Shalev-Shwartz and Ben-David (2014).

Logistic regression relies on the sigmoid function $\phi_{sigmoid} : \mathbb{R} \to [0, 1]$ (Equation 4) applied over a linear combination of a feature vector $\boldsymbol{x}$ and the model weights $\boldsymbol{w}$. The weights can be learned from training data using standard methods like maximum likelihood estimation or gradient descent and the logistic loss function (Equation 5).

A learned model is defined as $h_{\boldsymbol{w}} = \{\boldsymbol{x} \to \phi_{sigmoid}(\boldsymbol{w}^T\boldsymbol{x}) : \boldsymbol{w} \in \mathbb{R}^d\}$. The model outputs the probability of a document having the target label, i.e. the model $h_{\boldsymbol{w}}$ gives the probability of an instance $\boldsymbol{x}$ having class label ($y = 1$). When $\boldsymbol{w}^T\boldsymbol{x}$ is large the probability is close to 1.

$$\phi_{sigmoid}(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

$$\ell(h_{\boldsymbol{w}}, (\boldsymbol{x}_i, y_i)) = \log(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}}) \tag{5}$$

We use a one versus all logistic regression ensemble as a comparison model in Chapter 4.

2.3.3 *Support Vector Machines*

SVMs are motivated by a geometric understanding of classification: a set of linearly separable instances from two classes can be correctly divided into two groups by many different decision boundaries. Decision boundaries that are closest to the mid point between the two groups should generalise to new data the best. SVMs use this intuition as a starting point to find a decision boundary that maximises the distance to instances on either side of the boundary. Boser, Guyon, and Vapnik (1992) introduced SVMs for solving linearly separable data sets: the hard-margin SVM.

Cortes and Vapnik (1995) enabled SVMs to handle non-linearly separable data by relaxing the hard-margin constraint. Relaxing the constraint allows the model to violate the margin and commit errors on training data. This is known as the soft-margin SVM and is described as $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geqslant 1 - \xi_i$ for all i. The slack variables ξ_i range over the entire data set and allow the margin between training instances and the decision boundary to be violated.

To learn the model weights a learning algorithm minimises the norm of the weight vector $\boldsymbol{w}$ and the average margin violation over the training data. The tradeoff between

minimising $\boldsymbol{w}$ and the margin violations is controlled by a hyper-parameter $C > 0$. This leads to the soft-margin objective function

$$\min_{\boldsymbol{w},b,\xi} \left[\frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \xi_i \right] \tag{6}$$

such that

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geqslant 1 - \xi_i, i = 1 \ldots N, \tag{7}$$

$$\xi_i \geqslant 0, i = 1 \ldots N \tag{8}$$

Shalev-Shwartz, Singer, et al. (2011) presented a fast sub-gradient descent method for minimising the SVM objective function. Wang and Manning (2012) showed that SVMs have competitive results in text classification.

2.3.4 *Weighted Support Vector Machines and SVM+*

This thesis deals with the issue of using the topical nature of language as prior knowledge for building classifier ensembles for document classification tasks. Lauer and Bloch (2008) give a survey of different methods for introducing prior knowledge into SVMs. We will look at the experimental findings in more detail in the second part of this Chapter. Here we will look at the mathematical formulation for incorporating prior knowledge into SVMs.

Lin and Wang (2002) introduce Fuzzy Support Vector Machine (FSVM) and redefine the soft-margin objective function to include per instance error terms:

$$\min_{\boldsymbol{w},b,\xi} \left[\frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{M} c_i \xi_i \right] \tag{9}$$

such that

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geqslant 1 - \xi_i, i = 1 \dots N, \tag{10}$$

$$\xi_i \geqslant 0, i = 1 \dots N \tag{11}$$

The work is motivated by wanting to emphasize some training instances and de-emphasize others. Some training instances are known to be more important to classify correctly, so it is natural to bias the learning algorithm to pay more attention to those important instances. The c_i terms in Equation 9 allow setting a variable penalty for each instance in the learning set D_l. In the non-weighted SVM setting these weights can simply be set to one $c_i = 1, i = 1 \dots M$ and only the overall penalty term C is optimised.

Vapnik and Vashist (2009) introduced the Learning Using Privileged Information learning paradigm for SVMs. In this paradigm the learning algorithm gets additional information about each training instance from a "*teacher*". The information supplied by the teacher is privileged in that it is only available at learning time; at test time the learned model operates as any other without the additional information.

Lapin, Hein, and Schiele (2014) relate the SVM+ algorithm of Vapnik and Vashist (2009) with weighted SVM learning and show that the information given to SVM+ as privileged features can also be encoded as instance weights for the weighted SVM. Lapin, Hein, and Schiele (2014) show how the instance weights can be learned in the absence of privileged features.

We use the weighted SVM in all experiments in Chapter 3 to build topical ensembles.

SECTION 2.4
Ensemble Models

In this Section we look at ensemble models and how they can be used to improve classification performance. The idea behind model ensembles is to learn a number of models (instead of just a single one) and aggregate model predictions. The most common aggregation is majority voting, i.e. to predict the class with the most votes among the models.

Ensemble models often improve performance by reducing the reliance on any one sample of training data. In a binary problem, if there are 15 models in an ensemble a

document will be misclassified only if at least 8 of the models vote for the incorrect class. If a single model makes an error due to some anomaly in its training data, the other models in the ensemble are unlikely to make that same error as they will have been trained with a different sample. In our work we use ensemble models that are specifically targeted to have a topical bias; each individual model in the ensemble has some topical knowledge that the other models in the ensemble do not have. The crucial point is to reduce correlations in the errors the models make. This Section highlights different ways in which this decoupling has been done in the past.

In the following discussion we will refer to the *base classifier* as the learning algorithm used to learn models in the ensemble. Model ensembles can also be built out of a heterogeneous mix of different base classifiers. Model, predictor and hypothesis are used interchangeably to refer to a fitted model.

2.4.1 *Boosting*

Boosting[9] is a general purpose ensemble technique that can be used with any learning algorithm that supports sample weights. The sample weights allow models to be fitted sequentially such that each consecutive model is trained on a reweighted training set. The weights for a new model are determined according to errors the previous model made. This allows models further down the hierarchy to pay more attention to the "difficult" examples in the training set. Different boosting algorithms differ in the way in which the weights for the data at each step of the algorithm are calculated.

Let h^m denote the model trained at step m, $h^m(\mathbf{x})$ denote the predictions made by model h^m and $p^m = p^m_h(\mathbf{x})$ denote a probability distribution over the predictions $h^m(\mathbf{x})$. The vector $\mathbf{p}^m$ is a probability distribution at step m over the instances. A boosting algorithm A incurs a loss $\ell(A)$ proportional to $\mathbf{p}^m$, that is $\ell(A) = \sum_{i=1}^{n} p_i^m \ell_i^m$ where ℓ_i^m is the loss suffered by model h^m on the i^{th} instance.

The boosting algorithm decides how to compute $\mathbf{p}^m$ at each step from the losses the current model(s) is making and therefore how to allocate the instance weights for the new model.

9 Boosting has been called arcing in some early work in this area (Breiman, 1997).

2.4.1.1 *Adaptive Boosting*

Adaptive Boosting (AdaBoost) is an adaptive boosting algorithm introduced by Freund and Schapire (1997) that, as the name suggests, adapts the instance weights based on the errors of the model at step m. AdaBoost relies on training instance weights that are set based on the errors from the previous model in the ensemble, and prediction weights that scale the predictions of each consecutive model.

Freund and Schapire (1997) initialise the weight vector to be all ones and $\mathbf{p}^0$ to be a uniform $1/N$ in the absence of any prior knowledge of the problem domain. For step m they first calculate the error $\epsilon^m = \sum_{i=1}^{N} p_i^m |h^m(\mathbf{x}_i) - y_i|$. The updated weights w^{m+1} are then set according to

$$w_i^{m+1} = w_i^m \beta_m^{1-|h^m(\mathbf{x}_i)-y_i|} \tag{12}$$

where β_m is defined as $\frac{\epsilon^m}{1-\epsilon^m}$.

The vector of probabilities is then updated according to

$$\mathbf{p}^{m+1} = \frac{\boldsymbol{w}^{m+1}}{\sum_{i=1}^{N} w_i^{m+1}} \tag{13}$$

The per instance learned instance weights for consecutive models in the ensemble are thus scaled based on the errors the previous model made. The predictions the current model makes are scaled proportional to the weights assigned to the learning weights that were used to train the model.

2.4.1.2 *Gradient Boosting*

Friedman (2001) and Mason et al. (1999) expanded the boosting paradigm to a gradient descent method allowing general purpose function optimisation. The insight is in formulating an ensemble F as a linear combination of functions $f \in \mathcal{F}$ (fitted models) each sampled from a class of functions $\mathcal{F}$. Boosting can be seen as a way of optimising any differentiable loss function over the set of all possible linear combinations $\mathrm{lin}\,\mathcal{F}$. New functions (models) f are added to the collection F based on which new model reduces a

cost function the most. Both Friedman (2001) and Mason et al. (1999) show that finding the f that maximally reduces the error of the ensemble is equivalent to fitting a new model on the error residuals of the existing ensemble.

2.4.2 *Bagging Ensembles*

Bootstrap aggregation (bagging) (Breiman, 1996a) is an ensemble method for generating multiple predictors and aggregating the votes of those predictors together to get a single model. The main contribution of Breiman (1996a) was showing that by averaging multiple predictors the lower bound on accuracy can be raised and performance therefore improved. The authors also suggested a way of training predictors that can be averaged.

The bagging ensemble model inspired many other versions of training the ensemble, the difference between them being the way in which data is sampled. The central question in Chapter 3 of this thesis is whether prior knowledge of the topical structure of a classification dataset can be used to guide the model sampling, instead of performing that sampling at random.

We will first cover the bagging ensemble and how averaging predictions raises the lower bound on accuracy before we highlight the differences between the models that extend bagging. Unlike boosting methods all models in the ensemble are independent and can be fitted in parallel.

2.4.2.1 *Averaging Predictors*

Breiman (1996a) shows that a collection of weak learners can improve classification accuracy over a single learner. Let $Q(j|\mathbf{x}) = P(h(\mathbf{x}; s') = j) : s' \in S$ denote the relative frequency of a predictor h for predicting class j at $\mathbf{x}$ over many independent samples s' from S. The number of samples is an empirical question and is undefined for the time being. $P(j|x)$ is the true probability that $\mathbf{x}$ has class label j.

The probability that h makes a correct classification at the i^{th} sample $\mathbf{x}_i$ is

$$p(h(\mathbf{x}_i) = y_i) = \sum_j Q(j|\mathbf{x}_i)P(j|\mathbf{x}_i). \tag{14}$$

Equation 14 is just the weighted sum over all the class labels, with the weight being the relative frequency of label j given a number of independent models. The overall probability of correct classifications is

$$r = \int \left[\sum_j Q(j|\mathbf{x}) P(j|\mathbf{x}) \right] P_X(d\mathbf{x}). \tag{15}$$

Equation 15 is a lower bound on the classification accuracy of the collection of models when their predictions are not averaged together but the predictions are made simply based on the relative predicted class frequencies for each individual example $\mathbf{x}$. It is worth noting that Equation 15 is never higher than $\max_j P(j|\mathbf{x})$, the true probability of label j for $\mathbf{x}$.

Let $h_{avg}(\mathbf{x}) = \arg\max_j Q(j|\mathbf{x})$ be the majority vote ensemble model. For this model the ensemble's prediction at $\mathbf{x}$ is the class with the highest frequency among all the model predictions in the ensemble. Following Breiman (1996a) the probability of correct classification at $\mathbf{x}_i$ for the ensemble is

$$p(h_{avg}(\mathbf{x}_i) = y_i) = \sum_j I(\arg\max_i Q(i|\mathbf{x}_i) = j) P(j|\mathbf{x}_i) \tag{16}$$

where $I(\cdot)$ is the indicator function. If $h(\mathbf{x})$ assigns the highest probability to the correct class for $\mathbf{x}$ then the left hand side in Equation 16 equals $\max_j P(j|\mathbf{x})$. Let C be the set of examples where h ranks the class labels according to their true probabilities, i.e. the relative frequencies of the class predictions follow that of their true probabilities, and let C' be the complement of C. The probability of correct classification overall for the collection h_{avg} is then

$$r_{avg} = \int_{\mathbf{x} \in C} \max_j P(j|\mathbf{x}) P_X(d\mathbf{x}) + \int_{\mathbf{x} \in C'} \left[\sum_j I(h_{bag}(\mathbf{x}) = j) P(j|\mathbf{x}) \right] P_X(d\mathbf{x}). \tag{17}$$

By averaging the predictions of the ensemble the lower bound on classification accuracy is raised provided that for most of the inputs h assigns the relative class frequencies in their true order. Breiman (1996a) makes the following note "*If a predictor is good in the*

sense that it is order-correct [classes correctly ranked] *for most inputs* $\mathbf{x}$*, then aggregation can transform it into a nearly optimal predictor.*" It is worth noting that the discussion revolves around a predictor that is good and what happens when many such predictors are averaged together, not around how the predictors should be trained.

2.4.2.2 *Bootstrap Aggregation*

In addition to showing that averaging predictors can be beneficial Breiman (1996a) proposes to use the bootstrap method (Efron and Tibshirani, 1994) for taking repeated samples with replacement from the labelled data to create training sets for models in the ensemble. Each sample is the same size as the original dataset.

Bootstrap resampling is a general method for estimating the parameters of estimators from an empirical distribution. In the case of sampling training and test data sets it has the benefit that each new draw is done independently and identically distributed (i.i.d.) from any previous draws. As the sampling is done with replacement the empirical distribution does not change between sample draws. The benefit is that for a large number of sample draws the frequency of any single data point over all the samples is roughly equal to the frequency of any other data point.

2.4.2.3 *Pasting*

Breiman (1999) introduced pasting as a way of training model ensembles from datasets that do not fit into memory. The method uses either uniform sampling (without replacement) to form a large number of small training sets from the original, or alternatively importance sampling in which instances where an out-of-bag classifier (Breiman, 1996b) commits errors are preferentially selected. The results show that the importance sampling method produces better results than pure random sampling.

2.4.2.4 *Random Subspaces*

While bagging and pasting focus on sampling the training instances Ho (1998) introduced a method for sampling features instead of instances. All training instances are then projected onto the small number of selected features per model, setting the value of all unselected features to 0. At test time new instances are also projected to the random subspace of each model and the predictions are aggregated together based on the

predicted conditional class probabilities. The authors show a significant improvement in performance over other decision tree based ensembles on five different datasets.

2.4.2.5 *Random Patches*

Louppe and Geurts (2012) combine Pasting with Random Subspaces proposing a method called random patches; their method samples both instances to train new models from as well as a subspace of features for the sampled instances. Although the method does not show an improvement in accuracy it does lower the computational costs of building the individual models while maintaining performance.

The ensemble methods described in this subsection (Boosting, Bagging, Pasting, Random Subspaces and Random Patches) are so called meta-algorithms, general purpose ensembling techniques that can be used with any base estimator or indeed a heterogeneous mixture of base estimators. To conclude the discussion on ensemble models the next subsection briefly covers two algorithms that are specifically related to decision trees.

2.4.2.6 *Random Forests and Extremely Randomized Trees*

Random Forests (Breiman, 2001) and Extremely Randomised Trees (Extra Trees) (Geurts, Ernst, and Wehenkel, 2006) are two ensemble methods specifically designed for decision trees.

Random Forests combine the idea of bagging (Breiman, 1996a) and Random Subspaces (Ho, 1998) and adapt the techniques to decision trees by modifying the tree growing algorithm by selecting a random subset of features at each node to split on. The split value of the random subset is computed using a metric such as information gain. The method is effective in correcting for overfitting to the training data, a problem that is common for single decision trees. The predictions of the ensemble are formed using majority voting, similar to all the other ensemble models.

The Extra Trees algorithm (Geurts, Ernst, and Wehenkel, 2006) extends Random Forests by randomising also the splitting criteria at each internal node of the tree. Given randomly selected features of randomly selected instances in the current training set, the split value for each feature under consideration is also randomised instead of computed using one of the common metrics (Gini, Information Gain).

SECTION 2.5

Topic Models

This Section is focussed on topic models, an unsupervised method for clustering words and documents according to word co-occurrence patterns. Topic modelling forms a central part of this thesis; it is used as the basis for building classifier ensembles in Chapters 3 and 4.

Various different instantiations of topic models have been proposed but they all have in common the notion that documents are a mixture of topics[10]. While documents are mixtures of topics, topics themselves are modelled as probability distributions over a vocabulary.

2.5.1 *Latent Semantic Analysis*

The precursor of topic models was originally developed as a means of performing dimensionality reduction that preserves information about the co-occurrences of words within a document. Dumais et al. (1988) and Deerwester et al. (1990) introduced LSA, which relies on performing Singular Value Decomposition (SVD) on a sparse document-term matrix. The work is focussed on reducing the dimensionality and sparsity of the document space without losing information about the semantic relatedness of terms.

Let M be an $m \times n$ document-term matrix denoting the term frequencies in each document with m documents and n terms. SVD is a matrix factorisation method that allows decomposing the sparse document term matrix into linearly independent factor matrices U, V and Σ of dimensionality $m \times m$ and $n \times n$ and $m \times n$ respectively. Σ is a diagonal matrix containing the singular values, and U and V contain orthonormal basis vectors on their columns for documents and terms. Many of the singular values in Σ are small and can be ignored, thus reducing the dimensionality of the representation without losing the expressiveness of the data. The number of dimensions kept is an external parameter that needs to be set.

Reducing the dimensions of the document-term matrix to contain fewer terms allowed Deerwester et al. (1990) to match or improve over a number of previous results in IR.

10 Very early work on LSA (Dumais et al., 1988; Deerwester et al., 1990) does not treat documents as a mixture of topics, but instead each document is assigned a single topic.

2.5.2 *Probabilistic Latent Semantic Analysis (pLSA)*

Hofmann (1999b) and Hofmann (1999a) extends the work on LSA by putting the factor analysis of document-term matrices in a probabilistic framework. Instead of factoring documents and terms into topics using SVD, probabilistic Latent Semantic Analysis (pLSA) treats documents and words as observed variables in a graphical model and relates them to each other via a latent topic, one per document (Figure 1).

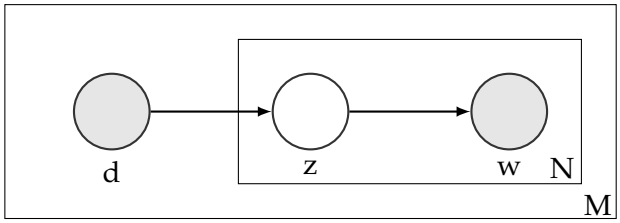


Figure 1: Plate diagram for Probabilistic Latent Semantic Indexing. Document labels d and words w are both treated as observed variables linked together via latent unobserved topic z. There is no natural way of determining the topic mixture z of unobserved documents as the topic mixtures are directly linked to data observed during training.

The probability model proposed by pLSA states that a document label and a word in the document are conditionally independent given some unobserved topic:

$$P(w, d) = P(d)P(w|d) \tag{18}$$

$$p(w|d) = \sum_{z \in Z} p(w|z)p(z|d). \tag{19}$$

The reasoning behind 18 and 19 according to Hofmann (1999b) is "*... observation pairs* (d, w) *are assumed to be generated independently; this essentially corresponds to the 'bag-of-words' approach ... the conditional independence assumption is made that conditioned on the latent class* z*, words* w *are generated independently of the specific document identity* d".

The inference of a pLSA model requires learning $kV + kM$ parameters, where k is the number of topics, M the number of documents and V the number of terms. The number of parameters therefore grows as the size of the text collection grows; in practice this leads to over-fitting (Popescul et al., 2001; Blei, Ng, and Jordan, 2003). Furthermore as the joint probability $P(d)P(w|d)$ depends on conditioning the topic mixture of documents on documents that have been observed the applicability to unseen data is limited: there

is no natural way in the model to determine the topic mixture of previously unseen data. This deficiency as well as issues with over fitting was addressed by Blei, Ng, and Jordan (2003).

2.5.3 *Latent Dirichlet Allocation*

Blei, Ng, and Jordan (2003) extended pLSA to a Bayesian generative model alleviating the problem of over fitting. While pLSA treats the word topic probability distributions as maximum likelihood estimates from data, Latent Dirichlet Allocation (LDA) treats them as parametrised distributions in a Bayesian framework: topics are multinomial distributions over the vocabulary and documents are multinomial distributions over topics (Figure 2).

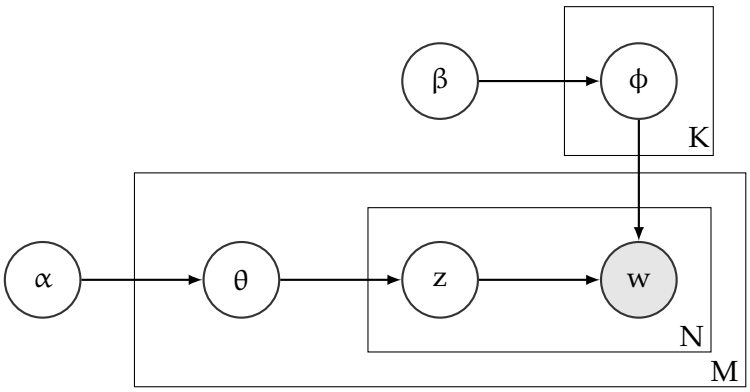


Figure 2: Plate diagram for Latent Dirichlet Allocation where only words are observed variables. The parameters α and β describe Dirichlet distributions that allow for sampling multinomial document-topic (θ) and word-topic (ϕ) distributions. The words w are each sampled from a separate topic distribution.

In LDA the document-topic distribution of each document is a random variable $\theta \in \mathbb{R}^k$ sampled from a Dirichlet distribution with parameters $\boldsymbol{\alpha}$

$$p(\theta|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{k} \boldsymbol{\alpha}_i)}{\prod_{i=1}^{k} \Gamma(\boldsymbol{\alpha}_i)} \prod_{i=1}^{k} \theta_k^{\boldsymbol{\alpha}_i - 1} \tag{20}$$

where Γ is defined as the gamma function (W. Olver et al., 2010).

The parameters $\boldsymbol{\alpha}$ can be thought of as prior observations of a topic with a document. The parameters are typically set to a uniform distribution although other options are

possible and may be preferable in some circumstances (Wallach, Mimno, and McCallum, 2009).

Given the document-topic distribution θ over k topics each word w_i in the document is regarded as having been generated by one of the k topics. The topics themselves are multinomial distributions over the vocabulary and have hyper-parameters β.

The probability model for a document can be described in terms of parameters α and β as

$$p(\boldsymbol{w}; \alpha, \beta) = \int p(\theta|\alpha) \left[\prod_{n=1}^{N} p(w_n|\theta, \beta) \right] d\theta. \tag{21}$$

This model although easy to relate to the probability model of pLSA (Equations 18 and 19) hides some detail as the topic assignments for each document are not explicit; expanding θ gives

$$p(\boldsymbol{w}; \alpha, \beta) = \int p(\theta|\alpha) \left[\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right] d\theta. \tag{22}$$

Exact inference of model parameters is not possible, but multiple approximate inference methods have been proposed including variational Bayes and Gibbs sampling (Blei, Ng, and Jordan, 2003; Griffiths and Steyvers, 2004; Hoffman, Blei, and Bach, 2010).

Multiple extensions to LDA have been proposed, among them are models that include a temporal aspect to LDA such as Topics Over Time (Wang and McCallum, 2006) and Dynamic Topic Models (Blei and Lafferty, 2006), Pachinko Allocation which aims to capture correlations between topics (Li and McCallum, 2006) as well as structured topic models (Wallach, 2006) that make stronger assumptions about word order than the standard bag-of-words assumption. LDA has also been extended by relaxing the need to define a fixed a number of topics (Teh et al., 2006) as well as using different parametrisations of the word-topic distributions and non-parametric models that utilise word embeddings (Das, Zaheer, and Dyer, 2015; Batmanghelich et al., 2016).

More closely related to the work presented in this thesis are the extensions to LDA that add the possibility of learning document labels, known as supervised topic models (Blei and McAuliffe, 2007; Zhu, Ahmed, and Xing, 2012; Li, Ouyang, and Zhou, 2015; Jameel,

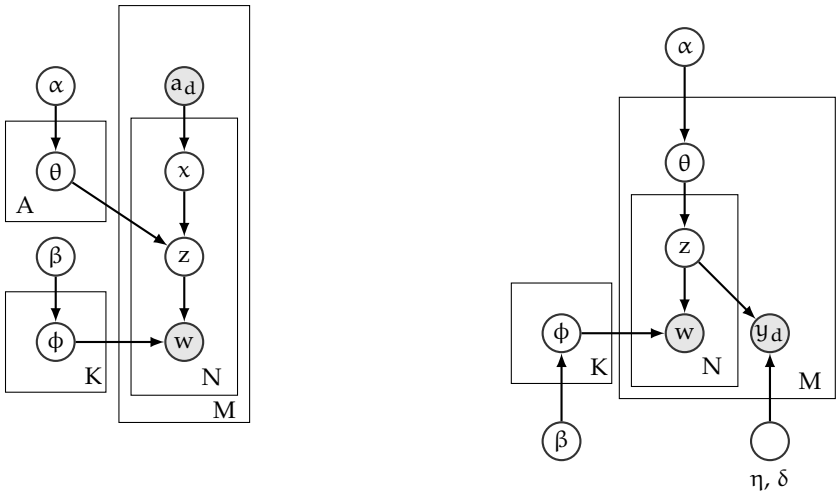
Lam, and Bing, 2015; Lacoste-Julien, Sha, and Jordan, 2008). The next two subsections cover supervised extensions to LDA and the use of LDA in supervised classification tasks.

2.5.3.1 *Supervised Extensions to LDA*

Rosen-Zvi et al. (2004) extended LDA to the Author-Topic model (Figure 3a), to allow learning not only the co-occurrence of words within documents but also the co-occurrence of discreet document labels, in this case an author identifier. The model adds a separate target label for each document determining which authors wrote the document. The document has multiple document-topic distributions each one conditioned on an author. Each author has unique mixture weights for topics describing which topics the author tends to write about. The authors of a document are not treated as a random variable.

Blei and McAuliffe (2007) presented supervised Latent Dirichlet Allocation (sLDA) and provided a more general purpose formalisation of supervised topic models than that of Rosen-Zvi et al. (2004). The authors formalised supervised topic models as a generative process that samples a target label, or "response variable", from the topic distribution of each document (Figure 3b). The response variable is not necessarily a multinomial but a generalised linear model (McCullagh and Nelder, 1989) which allows flexibility in the type of response variable used. The authors show that Gaussian and Poisson distributed response variables have an exact solution and develop a general-purpose approximation method for response variables that follow other distributions.

Lacoste-Julien, Sha, and Jordan (2008) presented a different formalisation of supervised LDA models called Discriminative Latent Dirichlet Allocation (DiscLDA) (Figure 4). In their model the document topic distribution θ is not sampled directly from a Dirichlet distribution but from a transformed mixture of Dirichlet distributions where the mixture weights determine the contribution of each target label to the document topic distribution. Their work turns the model of Blei and McAuliffe (2007) on its head. Instead of sampling a response variable from the topic distribution of each document, the document-topic distribution is sampled from a transformed Dirichlet space, where the transformation is label specific. The aim is to have the document-topic distribution of documents that belong to the same label resemble one another.



(a) Author-Topic Model

(b) Supervised LDA (sLDA)

Figure 3: Author-Topic model (Rosen-Zvi et al., 2004) and Supervised LDA (sLDA) (Blei and McAuliffe, 2007)

Zhu, Ahmed, and Xing (2012) explicitly pointed out the difference between DiscLDA and sLDA, the former being an upstream supervised LDA model and the latter a downstream supervised LDA model. A downstream model is one where the target label is predicted (generated by) the latent document representations – the document topic distribution – whereas in an upstream model the document representation is conditioned on the target label. These two views on how to incorporate the target label are conceptually very different although lead to similar results in terms of classification performance. The authors also present their model Maximum entropy discrimination LDA (MedLDA). The MedLDA model combines a discriminative classifier with a generative model that does not necessarily need to be LDA. The authors describe a framework for jointly optimising the parameters of the generative model and the discriminative classifier under a constrained optimisation framework. The framework aims to find a trade off between minimising the negative log likelihood of unlabelled data for the generative model on one hand and the prediction error of the discriminative classifier on the other. Variational Bayes methods are used to estimate model parameters, Zhu, Chen, et al. (2013) later proposed Markov Chain Monte Carlo (MCMC) methods, that improve both time efficiency and classifier performance for estimating the parameters.

The work of Zhu, Ahmed, and Xing (2012) is interesting as it explicitly acknowledges that the latent topical representation achieved by maximising the log likelihood of unlabelled documents is not necessarily appropriate for classification tasks. They also have

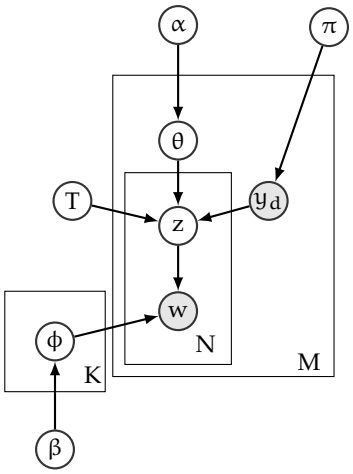


Figure 4: Discriminative LDA. Notice that the causality between the document-topic distribution (z) and the response variable y is reversed compared to sLDA (Figure 3b). π is a prior distribution for the response variable and T is a linear transformation matrix learned from data using Expectation Maximisation. The learned matrix is applied to document-topic distributions to allow the topics to discriminate between different target labels.

an informative discussion about when one should consider using MedLDA compared to a SVM classifier. Our work presented in Chapters 3 and 4 is similar but looks at the issue specifically from the point of view of how the topical information can be used to aid classification, not how the target labels can be used to modify the topic model parameters.

Ramage et al. (2009) extend the standard LDA model to include known document labels. The label set of a corpus is mapped one-to-one to the topics of the model, that is, each topic is constrained to correspond to a single label. Document labels are sampled from a Bernoulli distribution and these labels restrict the topics that are considered for sampling the words in the document. Our model presented in Chapter 4 is more flexible as it allows the document labels to distribute across a number of topics.

Soleimani and Miller (2016) present a semi-supervised multi-label topic model (Multi Label Topic Model (MLTM)). Under their model each topic is associated with a label distribution – not a single label – and each sentence in a document is labelled with a possibly empty set of target labels. The target labels for a sentence are sampled separately for each word according to the label distribution associated with the topic that generated the word in question. This model allows the target labels to distribute across all topics similar to our work in Chapter 4. The downside of their model is that adding new labels requires the entire model to be relearned, whereas our model can keep the

underlying topic model fixed, and only learn the weights between a new label and the already existing topics. The MLTM model is extended by Soleimani and Miller (2017) to handle structured data.

2.5.3.2 *Topic Models in Classification Tasks*

Topic models infer a set of latent topics from a corpus. Each topic in a topic model is a probability distribution over the vocabulary of the corpus. Given the topics, documents can be decomposed into a low-dimensional topic vector. This representation of the document contents has been used in several supervised classification scenarios as a replacement for the sparse high-dimensional document-term co-occurrence matrix. In this sub-section we cover the relevant parts of that literature.

One popular method for utilising topic models in classification is to use the topic model as a dimensionality reduction technique. This technique relies on the topic model's ability to produce a low dimensional representation of documents that capture at least some of the document's semantics. Chen and Hsieh (2006) apply this methodology to web page retrieval using LSA extracted features along with other features as input to an SVM classifier. Zhong and Zou (2011) also use Principal Components Analysis (PCA) for dimensionality reduction in an ensemble setting of SVMs for web page classification. These approaches utilise the topic model as a dimensionality reduction technique; our work is different in that we fold in the topical information in the construction of the ensemble as opposed to it being the signal to the models.

Mei et al. (2007) present a topic sentiment model to capture relationships between topics and sentiment. The standard LDA mixture model is extended by adding sentiment specific mixture components. The positive and negative mixture components (multinomials) are learned separately and impact the standard k topics of LDA via weight parameters that are learned from data or set using prior knowledge. The authors make the same argument as we do that sentiment and topicality are two separate pieces of information and one distributes over the other; we additionally argue that the distributive property depends on what the class labels are.

Lin and He (2009) proposed a joint topic sentiment model based on LDA. The standard LDA model is extended by adding a document specific sentiment distribution over a fixed number of sentiment labels. The sentiment labels condition the document-topic

distribution of each topic. The method is tested on movie reviews and has comparable performance with standard models such as SVMs.

Xiang and Zhou (2014) use LDA to improve sentiment classification in Twitter by building a topic based ensemble classifier. The ensemble is built by setting a threshold value and sub-sampling the training corpus according to that threshold for each topic; a separate SVM is built for each topic and majority voting is used to produce the final ensemble predictions. Our method discussed in Chapter 3 is similar to that of Xiang and Zhou (2014) in that a topic model is used in building the ensemble, however, we use the topical information directly in the loss function of the SVM and we explore a richer combination of prediction methods.

Van Canneyt, Claeys, and Dhoedt (2015) used a topic dependent classification model for sentiment classification on Twitter. Their model uses a universal model trained on all available data and a topic specific classifier together to perform classification. The training data for the topic specific classifier is sampled using hashtags as a topic proxy. The hashtags are clustered using spectral clustering; each resulting cluster represents a topic. Any document that contains any of the hashtags in a cluster is regarded as being part of that topic.

Chang, Ratinov, et al. (2008) introduced "Dataless Classification" as a framework where the target labels are not treated as meaningless indicator variables but as semantically meaningful labels. The label semantics are learned from world knowledge such as Wikipedia[11] articles. Unseen documents are then embedded in this same semantic space to perform categorisation. Although the authors do not use LDA to determine the semantic representation of documents, later modifications of the framework do. Chang, Ratinov, et al. (2008) learn explicit concept vectors for Wikipedia concepts[12] and use those vectors to perform classification on an unrelated dataset such as 20 Newsgroups[13]. The method essentially aims to find correspondences between the Wikipedia pages and the target domain for categorisation using a nearest neighbours classifier. The dataless classification paradigm was later expanded by Song and Roth (2014) to hierarchical problems and Chen, Xia, et al. (2015) developed a model based on LDA to compute the semantic representations of documents and target labels.

11 https://www.wikipedia.org
12 The concepts translate to page headers in Wikipedia
13 http://qwone.com/~jason/20Newsgroups/

The dataless classification paradigm highlights a key point made in this thesis, that document classification tasks can be divided into two groups based on the relationship between the target labels and the document content. All of the algorithms in the dataless classification literature are tested on tasks that require semantic labelling, but the utility of the semantic representation for tasks that require understanding the propositional content of a document is never tested.

Both Wang, Chen, and Liu (2016) and Shams and Baraani-Dastjerdi (2017) approach the task of aspect extraction and classification. This sub-task of sentiment classification is concerned with extracting aspect keywords under topical constraints; usually the target domain is product reviews. The research shares a common theme with the work in this thesis by highlighting the differences between topicality and features that are important for classification.

2.5.3.3 *Evaluating Topic Models*

In addition to several extensions to LDA itself, there is a growing body of work for ways to evaluate the quality of a trained model. As LDA is an unsupervised algorithm it can be difficult to evaluate if a model is fit for purpose, or if the hyper-parameters have been set in the best way. A central question is the number of topics that should be inferred.

Perplexity has been frequently used to compare trained topic models as well as for monitoring convergence during learning. Perplexity is problematic though, as it measures how well a probability distribution predicts unseen data. Predicting unseen data well does not mean that the topic model has learnt a good representation of the topical structure of an unseen corpus (Chang, Boyd-Graber, et al., 2009).

Essentially, the question boils down to what a "good topic model" is. Early approaches used perplexity, but Chang, Boyd-Graber, et al. (2009) showed this to be a bad metric as it does not produce models that correspond to human intuition about the topical relatedness of words, or of topics themselves. This led to research looking for ways to capture the internal coherence of the topics themselves as well as the coherence of the topics w.r.t. an external reference corpus.

Alsumait et al. (2009) focussed on the problem of automatically identifying insignificant and junk topics. Usually a trained topic model has some number of junk topics that contain idiosyncratic word combinations and insignificant topics that model background

word distributions. Mimno and Blei (2011) use Bayesian model checking methods to verify that the model has learnt what is important for the experimenter. Their framework is formalised as posterior predictive checks which measure aspects of the model posterior. The problem with this framework is that it requires the experimenter to define what to check for and is in some sense circular in that the checks easily end up checking assumptions the topic model itself is built on.

Lau, Newman, and Baldwin (2014) developed automated ways of evaluating the coherence of topics. Their methods rely on human evaluation of so called intruder words within topics, the intuition being that intruding words will be easy to detect from topics that are coherent but hard to detect from topics that are incoherent.

Recently Morstatter and Liu (2016) proposed new topic model evaluation measures that extend the work of Lau, Newman, and Baldwin (2014). Specifically, the authors are interested in measuring the coherence and interpretability of automatically inferred topics.

Röder, Both, and Hinneburg (2015) synthesised most of the existing research on topic model evaluation techniques and proposed a unified pipeline that can produce any one of the previously proposed methods, depending on how parameters are set. The pipeline contains four stages that compute the coherence of a trained model based on an external reference corpus.

Finally, an exception to the rule that evaluating topic models is difficult are applications where an external task can be used as a quality metric. The work presented in this thesis uses a supervised classification task to measure the quality of the whole classification pipeline, of which the topic model is a part. We do not, therefore, utilise the evaluation methods presented above.

SECTION 2.6
Multi-label Learning

In this final Section we review multi-label classification and the typical modifications that allow standard algorithms to handle the multi-label scenario; while some of the algorithms presented above can be used in multi-label settings many of them cannot without modifications. In multi-label classification the set of target labels is not mutually exclusive, and any number of labels can be applied to an instance. We focus on the main

approaches to multi-label classification in this Section. Comprehensive reviews on multi-label classification were recently published by Madjarov et al. (2012) and Galindo and Ventura (2014).

People are faced with a wealth of information that can easily become overwhelming. Digital services can help to reduce the information overload and allow users to focus on information that is interesting or relevant by sorting documents into category structures. Using such a category structure to label documents is known as Multi-Label Document Classification. In this task each document is labelled with n of k labels where n is unknown and can change from document to document. Typically, n is also significantly less than k. Examples of this application scenario include categorising news (Schapire and Singer, 2000; Katakis, Tsoumakas, and Vlahavas, 2008), legal text (Loza Mencía and Fürnkranz, 2008) and indexing medical research (Kakadiaris et al., 2016).

Often, in multi-label document classification scenarios, the category structure is hierarchical and the categories overlap with each other. Furthermore the categories often form a hierarchy. For instance, a top level label such as Sport would have sub-labels Tennis, Football, Icehockey and so on. The category structure can potentially be used as a rich source of information on how the labels relate to one another, allowing related labels to "borrow" information from each other in a learned model (Levatić, Kocev, and Džeroski, 2015; Cerri, Barros, and Carvalho, 2014). In some cases the category hierarchy is explicit[14] or it can be implicit, such as those in user created taxonomies (Tsoumakas, Katakis, and Vlahavas, 2008).

2.6.1 *Label Set Modifications*

A very simple meta algorithm for multi-label classification that can utilise any binary classifier is the One versus Rest (OvR) classifier ensemble[15]. This model trains a binary classifier for every label $s \in S$ using all documents that have that label assigned to them as the learning set for the target class and all documents that do not have the label as the learning set for the other class. Each trained classifier will decide whether

14 One example of an explicit category structure is the Reuters Corpus Version 1 introduced by Lewis et al. (2004). The dataset consists of approximately 800000 documents each labelled with a number of topical category identifiers.

15 The OvR strategy is also known as Binary Relevance.

their target class should be applied to an unseen test document. The method is efficient as the number of classifiers needed is equal to the number of classes in the label set, but is unable to take advantage of the label hierarchy or correlations between labels. Depending on the application scenario this may not be a problem.

Boutell et al. (2004) propose a number of modifications of the OvR strategy. One proposal is to ignore, either randomly or using a heuristic, all but one of the labels for all instances that have multiple labels, leaving each instance with just a single label and turning the problem as a whole into a multi-class one. Alternatively one could ignore all the multi-label instances in the dataset and only use instances that have a single label; depending on the dataset this technique can result in a significant reduction in the amount of data available for training, which can have a serious adverse impact on performance. Finally each combination of multiple labels can be considered itself to be a new label; a document with the labels *Sports* and *Tennis* would get a new label *Sports+Tennis*. This approach also would likely suffer from lack of data in many practical applications as the number of label combinations is of course much larger than the set of all labels while the number of labelled training instances remains unchanged. Documents labelled with *Sports+Tennis* would no longer belong to the training sets of either the *Sports* or *Tennis* labels but exclusively for *Sports+Tennis*.

Tsoumakas, Katakis, and Vlahavas (2008) present a label transformation method that builds a tree of labels from the complete label set. A multi-label classifier is trained for each internal node of the tree, using the union of labels in each sub-tree as the target labels. Classifiers higher up the tree can be thought of as splitting the label set into coarse blocks, whereas lower down the tree classifier become ever more refined and distinctive.

Li, Miao, and Pedrycz (2017) present a feature selection and label transformation method that takes label correlations into account. First the labels are clustered using k-means clustering, and then a feature selection method based on mutual information is applied separately to each cluster of labels. The method is tested using the ML-KNN classifier on a number of different data sets.

2.6.2 *Algorithm Modifications*

McCallum (1999) present a generative mixture model for multi-label classification that is similar to topic models. Each label c_j is represented by a distribution over the vocabulary $P(\boldsymbol{w}|c_j) : w \in V$. Each document is generated by a mixture of these labels according to mixture weights $\boldsymbol{\lambda}$.

Schapire and Singer (2000) applied the boosting framework to categorise news text into 93 different non mutually exclusive categories. They tested two different boosting algorithms each with a different loss function on the problem and showed their boosting algorithms to be best in class against a number of competitive methods. They also showed that the problem overall becomes more challenging as the number of classes increases: the error rate of the best performing model (theirs) doubled as the number of classes doubled. Al-Salemi, Aziz, and Noah (2015) used LDA with the same boosting algorithm as Schapire and Singer (2000) and showed that representing the documents as topic vectors instead of bag-of-words vectors not only decreases the computational cost but also increases performance.

Zhang and Zhou (2007) present a modification of the KNN classifier for multi-label data. Unseen test instances are assigned labels based on the label sets of the k nearest neighbours based on the maximum a-posteriori probability of the label set. This approach however is problematic for text classification as the original clustering of documents in the KNN algorithm is normally done using Euclidean distance, which has been shown to compress the dynamic range of similarity values as the dimensionality of the feature space increases (Aggarwal, Hinneburg, and Keim, 2001). In practice this means that the mean similarity of all documents increases as the number of features increases. Indeed the text categorisation data they use is aggressively downsized cutting out 98% of the feature space.

Tsoumakas and Vlahavas (2007) present the Random K Labelsets (RAKEL) algorithm which aims to utilise label correlations during training. Their method builds on the concept presented by Boutell et al. (2004) of using each distinct label combination as a separate label, called the label powerset method. The method works by selecting randomly without replacement k labels from the complete set of labels and then building a binary classifier for each label in that subset of labels. This process is repeated multiple

times to build an ensemble of multi-label classifiers. Label prediction for new instances is complicated, and involves each classifier making a binary decision for each label in the label set the classifier was trained with, then aggregating those decisions over the entire ensemble and comparing the average per label votes to a user specified threshold.

Rubin et al. (2012) showed that generative topic models can achieve competitive results in multi-label document categorisation compared to discriminative models. They present three variants of supervised LDA models (Flat-LDA, Prior-LDA and Dependency-LDA) that learn word-topic distributions in the context of category labels. Flat-LDA has the same parametrisation as the Author-Topic model of Rosen-Zvi et al. (2004) but uses a different data source for learning the document response variable. Prior-LDA extends Flat-LDA by taking corpus wide label probabilities into account and Dependency-LDA further extends that by also taking label correlations into account. The correlations are modelled using another topic model stacked on top of the one modelling word-topic correlations. The models show competitive results against standard SVM OvR ensembles especially when label correlations are taken into account (Dependency-LDA). Our results in Chapter 4 are in line with their findings.

Li, Ouyang, and Zhou (2015) present two models similar to those of Rubin et al. (2012): Frequency LDA (FLDA) and Dependency Frequency LDA (DFLDA). FLDA is designed to address limited training data and unlike Prior-LDA does not treat the document label as a random variable sampled from a multinomial but as an observed variable whose distribution is estimated from training data in maximum a-posteriori fashion. In DFLDA the document label and topic are coupled together allowing the generative process to 'share' latent topics that frequently co-occur under co-occurring document labels. The results for both FLDA and DFLDA are competitive compared to discriminative classifiers and in line with our results in Chapter 4.

Cerri, Barros, and Carvalho (2014) introduce local Multilayer Perceptron (MLP) for hierarchical multi-label problems. In their model each level in the label hierarchy gets its own multi-label classifier. The classifiers are trained sequentially using the predictions from the previous layer as the input to the next layer. The approach is motivated by deep learning architectures where consecutive layers in the network get as input the output of the previous layer.

Levatić, Kocev, and Džeroski (2015) perform an analysis of eight different datasets that have a hierarchical multi-label class structure. The research aims to explore the relationship between the label hierarchy and different classification models. They find that while single decision trees do benefit from the label hierarchy information, ensembles of decision trees do not. In Chapter 4 we show that the label hierarchy can be beneficial for ensemble models in text classification tasks.

Deep learning methods have in recent years become popular in many classification tasks. Multi-label classification is no exception. Liu et al. (2017) applied a deep CNN to a problem they call Extreme Multi-label Text Classification. The classification is extreme in the sense that the number of possible target labels is very high. In their experiments the number of labels for the data sets ranged from 103 labels to 670,000. They apply a standard CNN for text classification to the problem with one memory efficiency modification that reduces the number of parameters that need to be kept in memory during learning. In line with previous research they show that as the number of labels increases the performance of the algorithm decreases. However, their method is not only able to scale to all of the data sets, which the best performing comparison methods are not able to do, but they also beat the state-of-art on five of the six data sets.

2.6.3 *Label Dependence*

One shortcoming of the commonly used OvR strategy is its insensitivity to label correlations . Dembczynski and Cheng (2010) highlighted two different kinds of label dependencies, conditional and marginal. Marginal, or unconditional, label dependencies are simply those label correlations that are observed overall in the training data set. They are unconditional in the sense that the dependence or correlation between the labels is not conditioned on any specific learning instance. Conditional label dependence on the other hand is specifically related to an observed instance and relates to the loss function a learning algorithm uses. Dembczynski and Cheng (2010) also analysed a number of multi-label algorithms in how they address the label dependency issue.

To capture label dependencies Cheng and Hüllermeier (2009) combines KNN with logistic regression in a stacking model. They use the labels of neighbouring instances, as determined by the KNN classifier, as input features to a logistic regression model. This al-

lows the authors to integrate information about the likelihood of labels for neighbouring instances in the logistic regression learning process. Their results show that accounting for label dependencies in the local neighbourhood improves performance on a number of different data sets. Furthermore, the benefit of their specific method is that the degree to which certain label pairs are dependent on each other is directly encoded in the regression coefficients of their model.

Read et al. (2011) extended the Binary Relevance method to account for label dependencies. Their reasoning for using Binary Relevance is its linear scaling with respect to the size of the label space[16]. They train a chain of binary classifiers where the feature space of each consecutive classifier in the chain is extended with the true binary labels of all the previous models in the chain for the training data. This way, the classifiers later in the chain learn to model specific binary labels as combinations of previous binary labels and the input instances. This process is obviously sensitive to the order in which the classifiers in the chain are learned, and thus also the ordering of the labels. Read et al. (2011) therefore also present an ensemble of classifier chains, where each classifier chain is trained on randomly ordered labels. Their results indicate that although the ensemble of classifier chains shows competitive results, they come at a high computational cost. Further, one interesting aspect that the authors did not explore is the effect of label hierarchies on the ensemble classifier chain model. It would make sense for instance to learn the consecutive classifiers based on the hierarchy structure of the labels first learning the more general labels at the top of the hierarchy and then learning ever more specific labels further down the hierarchy.

Recently Zhang, Wang, et al. (2018) proposed a deep learning method that explicitly accounts for label dependence. They apply a feed-forward neural network to the problem, but embed both the documents and the hierarchical label space into low-dimensional spaces. The low dimensional document and label representations are combined during learning in a single loss function. Embedding the labels seems to help as their results indicate a significant improvement over state-of-the art methods.

16 While Binary Relevance scales linearly w.r.t. label space, methods such as the label powerset scale quadratically (Boutell et al., 2004; Tsoumakas and Katakis, 2007). This can be extremely problematic in multi-label classification as the number of categories for many data sets reaches into the thousands.

2.6.4 *Multi-label Classification in Other Domains*

While the algorithms presented above are mostly tested on news text, there are many other multi-label text categorisation tasks. A recent workshop focused exclusively on the problem of multi-label classification of medical research (Kakadiaris et al., 2016). The problem is one of assigning relevant keywords to medical research papers so that those papers can be found easily. The keywords come from a predefined set of classes, the Medical Subject Headings (MeSH)[17], organised in a hierarchy with lower levels of the hierarchy representing more specific information. The approaches to the BioASQ workshop range from using Elastic Search[18] to create semantic indices (Segura-Bedmar and Carruana, 2016), applying ranking algorithms (Zavorin, 2016) to model ensembles (Papagiannopoulou et al., 2016). While the standard Elastic Search indices had high recall but low precision, the learning to rank method performed well and increased performance over the baseline MTI system (Mork, Jimeno-Yepes, and Aronson, 2013; Mork, Aronson, and Demner-Fushman, 2017) which uses a combination of 3rd party services and machine learning models.

Similar to the MeSH classification hierarchy of medical research, the EUR-Lex[19] data set contains documents about European Union law. The data set contains different kinds of documents from treaties to case law and legislative proposals and is indexed by nearly 4000 different categories. Loza Mencía and Fürnkranz (2008) tested three different multi-label algorithms on the EUR-Lex data set. The methods they tested were: Binary Relevance, Multilabel Multiclass Perceptron and Multilabel Pairwise Perceptron. One issue highlighted by the research is the high computational cost of the Multilabel Pairwise Perceptron; on the roughly 4000 labels of the EUR-Lex data set 8000000 pairwise classifiers would need to be trained. To address this problem the authors develop a dual formulation of the multilabel pairwise perceptron algorithm that reduces the dependence on the label space. The algorithms are compare favourably to a relatively weak baseline model: multi-label Naï Bayes.

Finally, multi-label classification is not limited to the text domain. Levatić, Kocev, and Džeroski (2015) give the examples of gene function prediction and ecological community

17 https://www.nlm.nih.gov/mesh/
18 https://www.elastic.co/
19 http://eur-lex.europa.eu/

structure where multi-label classification problems also rise. Specifically, they focus on the importance of the label hierarchy in eight different multi-label classification data sets. Their results indicate that accounting for the hierarchical structure of the label space improves performance on all the test data sets.

Part II

TOPICAL ENSEMBLES

In part I we showed that there are many different kinds of document classification tasks that, in order to be performed effectively, require different signals: sentiment analysis requires understanding the propositional content of documents while document categorisation requires understanding the topical content. These tasks vary in difficulty, and in how the target labels behave with respect to the document content. However, the classification signal provided by the document content can be perturbed by changes in topical context.

While supervised extensions to topic models and other novel classification algorithms have been applied to document classification, the methods are often tested on tasks that require understanding topical, not propositional, content. In the second part we develop two classifier ensembles based on topic models, and we test the ensembles on document level sentiment analysis and hierarchical news categorisation tasks.

3

TOPICAL ENSEMBLES FOR SENTIMENT CLASSIFICATION

In this Chapter we look at the task of document level sentiment classification and how topical diversity affects classifiers at this task. Individual classifiers can struggle with topically diverse corpora due to document features changing their class association between topics. We show empirically that topical diversity does cause problems for single classifiers and develop an ensemble classifier that is sensitive to topical context. The aim is to improve classification performance in topically rich corpora by extending traditional discriminative models using a topically biased ensemble.

Consider a document classification task that is performed on a topically rich corpus, for instance, sentiment analysis of product reviews on a corpus that contains reviews from different product categories. A machine learning model trained to differentiate positive from negative sentiment needs to learn a mapping from document features to target labels. The mapping has to remain consistent over the entire application domain of the model, but domain dependent inconsistencies have been observed in multiple domain adaptation studies (Zhang, Hu, et al., 2015; Turney, 2002; Bollegala, Weir, Carroll, and Ishizuka, 2010) and certain pairs of domains have been found to be more challenging than others. In domain adaptation, however, the source and target domains are known and training data usually only exists for the source domain; we do not make these assumptions.

The key aspect to consider in the context of this Chapter is the class association of document features, and changes in that association. Since machine learning models infer a mapping that is based on observed data and we know from previous research that domain dependent changes exist, our hypothesis is that corpora with rich topical structure complicate learning a classification model. The question we explore in this Chapter

is if allowing the classification model to be aware of the topical context of documents improves performance at a sentiment classification task.

We will now present the topical ensemble used in the rest of this Chapter. The building blocks for the ensemble are Support Vector Machines (SVM), Latent Dirichlet Allocation (LDA) and ensemble models. For a description of the building blocks please refer to Chapter 2.

SECTION 3.1
A Topical Ensemble

So far we have described how changes in topical context can cause problems for single classifiers and outlined the building blocks for a topical ensemble model. In this Section we present the topical ensemble.

The topics inferred from natural language documents can be utilised to build an ensemble for text classification, creating a mixture of topical experts[1] instead of a collection of randomly sampled models. Each local model should capture unique aspects in the relation between topics, document features and the target labels. Previous research has shown topically biased ensembles to be effective in sentiment classification on Twitter data (Xiang and Zhou, 2014); we test our model on user generated product reviews.

The ensemble is designed to capture changes in feature-class associations across topics, by biasing each model in the ensemble to learn from documents that are closely related to a topic. We do not wish to find the relation between the topical composition of a document and the class labels, i.e. a mapping from the document-topic distribution of a document to the class labels. This approach has been used in early work exploring topic models as a dimensionality reduction technique (see Section 2.5 for a discussion). Instead, we want to find a mapping from the document contents to the target labels using the information derived from a topic model as a way to bias individual models in the ensemble to pay closer attention to specific topics.

We use a standard bag-of-words document representation for training the topic model as well as the ensemble. The classifiers learn to model the target classes from a bag-of-words document representation under a specific topical bias; the bias is created by

1 The term "mixture of experts" is used in the neural networks literature to refer to a specific ensemble model (Jacobs et al., 1991), we use the term in a general sense.

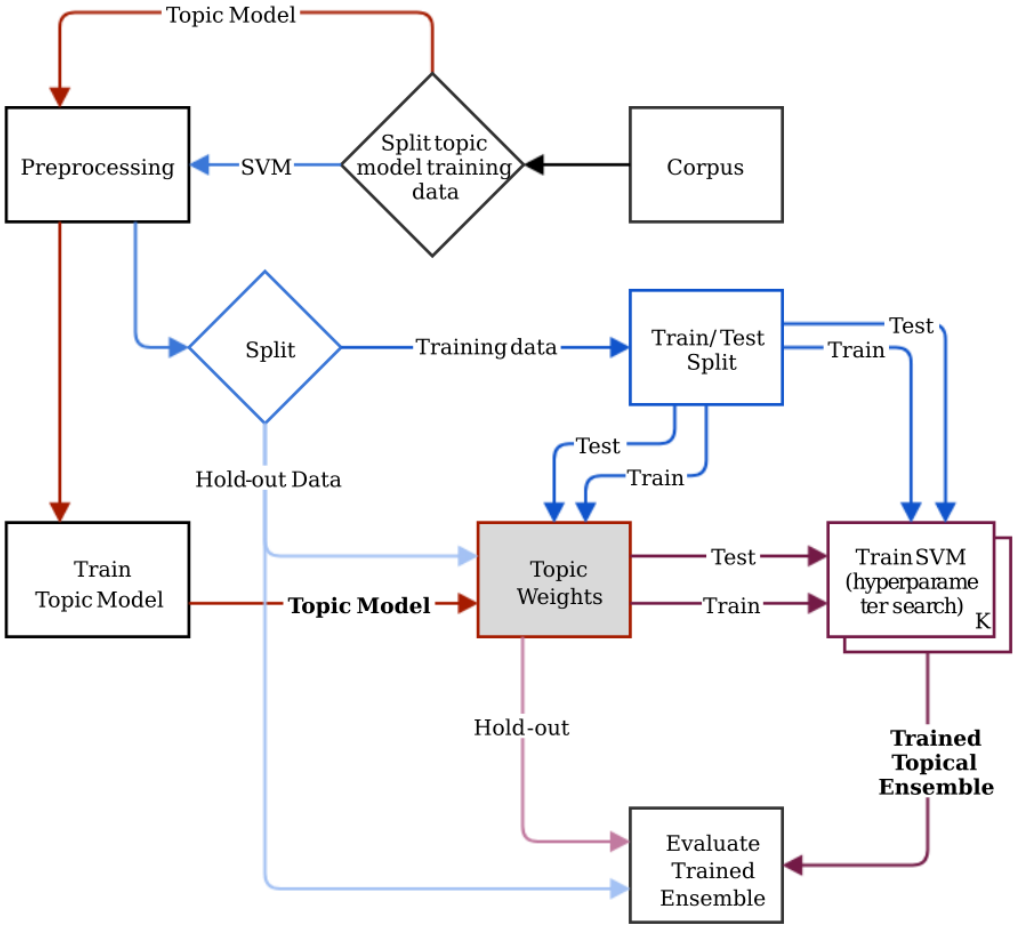


Figure 5: Training workflow for a topical ensemble. Notice that unlike in previous work we use the topic weights as additional input for the ensemble training, but they are not used as the document representation.

setting higher training weights to documents that are closely related to a topic. This allows the ensemble to model the content of documents under a certain topical bias.

Under the weighted SVM learning paradigm (Equation 9) each training instance can have a separate weight. The per instance weight is a variable penalty for misclassifying a training instance; setting the weights for in-topic documents high in relation to out-of-topic documents the SVM is forced to learn a model that on average makes fewer errors on the in-topic documents. We use the (scaled) topic weights $\theta_{ik} \in [0, 1]$ for the ith document and kth topic derived from LDA as the weights. This allows us to create a topically

weighted discriminative model; training k such models creates a topical ensemble. The kth SVM in the ensemble is given by the objective function

$$\min_{w,\xi} \left[\frac{1}{2}\|w\|_k^2 + C_k \sum_{i=1}^{M} \theta_{ik}\xi_i \right] \tag{23}$$

such that

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geqslant 1 - \xi_i, i = 1 \dots N, \tag{24}$$

$$\xi_i \geqslant 0, i = 1 \dots N \tag{25}$$

A crucial aspect of the training procedure is the scale of the instance weights θ_{ik}. Low overall weights will cause the learning algorithm to simply minimise the norm of the coefficient vector. This is a practical concern as the document-topic probability distribution typically contains only a few entries above 0.01. We will elaborate on the issue of scaling the learning weights in Section 3.5.2 later in this Chapter.

Ensemble predictions are produced either by unweighted majority voting or by taking the topic proportions of test data into account. Since each model in the ensemble is a topical expert, it makes sense to use the topical information not only during the learning phase but also in producing the final ensemble predictions. Weighted majority voting allows each classifier to participate in the prediction of the final label proportional to how well the classifier knows the topic of a document. We analyse the impact of different prediction methods in Sections 3.5.1.1 and 3.5.1.2.

Finally, we test a number of different parameters for the number of topics. Since the ensemble classifier as a whole is evaluated on a classification task, the quality of the topic model can be linked to the classification metrics. Topic models that increase performance at the classification task are better than those that do not.

3.1.1 *Closely Related Models*

Our work is similar to that of Xiang and Zhou (2014). They train a topical ensemble to perform sentiment analysis on social media data. Similar to our model they use the

topical information to inform training data selection for each topically biased model in the ensemble and use a bag-of-words document representation together with a number of hand crafted features to train the ensemble. The crucial difference between their model and ours is that we integrate the document-topic weights into the objective function of the models used in the ensemble. In their work the document-topic probabilities are used to create discrete sub-samples of training data. The closest comparable setting in our case is the unweighted majority voting ensemble (see Sections 3.4 and 3.5.1.1) although even in this setting our ensemble is trained with "soft" topical cluster assignments. Finally, their method is only tested on social media posts, but in Section 3.6.2.1 we compare our method of training the ensemble with theirs.

In Tables 1a and 1b we provide the results from a number of studies that also perform sentiment analysis on Amazon product reviews along with our results for a sample of the data sets we have used. Please note that the results in Tables 1a and 1b use different data samples and are not directly comparable with each other. The results in Table 1b are obtained by training our models with the same number of labelled instances as for the comparison results. However, as we do not perform domain adaptation the training sets for our method contain data from both of the marked categories. This is not the case for the results in Table 1a where the methods are tested on out-of-domain data. Finally, Table 1b also lists the results for the closest ensemble method (Xiang and Zhou, 2014) on our datasets.

	Accuracy			
	Book	DVD	Electronics	Kitchen
Support Vector Machine	0.70	0.70	0.75	0.77
SCL (Blitzer, Dredze, and Pereira, 2007)	0.76	0.76	0.82	0.85
SFA (Pan et al., 2010)	0.77	0.77	0.82	0.85
SST (Bollegala, Weir, and Carroll, 2011)	0.77	0.79	0.84	0.85
DSSC (Wu, Huang, and Yuan, 2017)	0.79	0.81	0.85	0.87

(a) Sentiment classification performance (accuracy) of other methods on Amazon product reviews. The metric reported on each column are obtained by training a model on the other three domain and testing the trained model on the domain the column is labelled with. The numbers reported are for reference only and are not comparable with the reported metrics from our method due to different data samples.

	Accuracy			
	I	II	III	IV
Support Vector Machine	0.87	0.88	0.87	0.86
Weighted Binary Ensemble (ours)	0.88	0.88	0.87	0.87
Topic Weighted Ensemble (ours)	0.87	0.88	0.87	0.86
Discrete Ensemble (Xiang and Zhou, 2014)	0.76	0.79	0.81	0.78

(b) Accuracy of our method on a selection of the category pairs and an even data sample (Table 3a). Our method is trained using the same number of training instances as the methods in Table 1a. The category pairs displayed in the Table are as follows:
– I *Books - Cell Phones and Accessories*
– II *Home & Kitchen - Office Products*
– III *Home & Kitchen - Baby*
– IV *Baby - Digital Music*

SECTION 3.2

Datasets - Amazon Product Reviews

In this Section we present the data sets we used in all our experiments. The data sets are samples from the Amazon Product Reviews corpus (He and McAuley, 2016)[2]. The original data set consists of approximately 80 million user submitted product reviews in 24 different product categories. Each review is associated with a sentiment score from 1 to 5 with respect to the product being reviewed, 1 being the lowest (most negative) and 5 being the highest (most positive). Although the sentiment score is associated with the product review, there are cases where the review and the user provided rating is about the seller or logistics company, not about the product itself. We have not estimated what

2 http://jmcauley.ucsd.edu/data/amazon/

proportion of reviews fall into this category. A category breakdown of the Amazon Product Reviews Corpus is given in Table 2; for most categories the positive reviews are the majority class. In the table we have rated reviews with 4 or more stars as positive, reviews with 2 or fewer stars as negative and anything in between as neutral.

Category	Negative	Neutral	Positive	Total
Books	2,099,427	1,924,880	18,525,677	22,549,984
Electronics	1,359,660	633,771	5,840,735	7,834,166
Clothing Shoes and Jewelry	750,635	574,871	4,426,948	5,752,454
Movies and TV	575,260	417,171	3,635,699	4,628,130
Home and Kitchen	661,374	345,484	3,253,323	4,260,181
CDs and Vinyl	295,122	266,605	3,216,567	3,778,294
Cell Phones and Accessories	755,501	351,749	2,345,472	3,452,722
Sports and Outdoors	417,457	278,392	2,575,768	3,271,617
Kindle Store	316,303	306,928	2,582,880	3,206,111
Health and Personal Care	464,804	242,051	2,281,689	2,988,544
Apps for Android	428,376	253,679	1,957,246	2,639,301
Toys and Games	308,986	194,057	1,751,464	2,254,507
Beauty	297,220	170,060	1,559,663	2,026,943
Tools and Home Improvement	280,696	153,309	1,494,197	1,928,202
Automotive	186,450	103,920	1,084,605	1,374,975
Video Games	230,848	124,589	972,388	1,327,825
Grocery and Gourmet Food	165,108	97,468	1,041,818	1,304,394
Office Products	230,733	99,287	915,352	1,245,372
Pet Supplies	192,524	105,072	940,930	1,238,526
Patio Lawn and Garden	177,285	80,932	736,067	994,284
Baby	133,450	83,019	700,300	916,769
Digital Music	50,124	41,548	747,967	839,639
Amazon Instant Video	60,444	41,243	484,165	585,852
Musical Instruments	57,624	38,576	404,587	500,787

Table 2: Document counts for each sentiment class for the 24 categories in the Amazon Product Reviews dataset. Negative documents have a sentiment rating $\leqslant 2$, positive documents $\geqslant 4$ and those in between are neutral.

To create the data sets for our experiments we first removed all documents with a sentiment score between 2 and 4, thus leaving only positive (score $\geqslant 4$) and negative (score $\leqslant 2$) reviews. Using this set of positive and negative reviews we divided the

categories into pairs and ranked all pairs according to a vocabulary agreement metric (see Section 3.2.1). The metric quantifies how well a feature's class association agrees with a target class across two topics. Section 3.2.1 describes the metric in detail.

We computed the agreement for all 276 topic pairs[3] and selected a representative sample of 8 pairs, 4 with high agreement and 4 with low agreement. The agreement metric was computed using all terms that occur in 5 or more documents in both categories; the probabilities were measured on the word level. The 8 category pairs were then sampled to create the final classification data sets. The aim is to test how the topical ensemble behaves when the distribution of topics and target classes changes. We therefore created six data splits, each 10000 documents in total, varying both the category and class balance in each. The splits are summarised in Table 3.

The splits were specifically selected to allow testing for differences in the performance of the topical ensemble compared to the baseline single SVM. As the single SVM explicitly models the class split of the data and only implicitly the category split of the data (through the coefficient vector), the topical ensemble should gain an advantage when the class balance starts to change between the categories (Tables 3c through 3f). In Section 3.6.1 we analyse in detail how the baseline method behaves in terms of the data splits.

3.2.1 *Vocabulary Agreement*

We now describe the vocabulary agreement metric we used to select the category pairs for the experiments. Vocabulary agreement is a measure for the difference in class association of a word type between two topics. In general, we would like the agreement measure to be positive when a word type's class association remains the same between topics, and we would like the measure to be negative if a word type's class association is different between topics.

We first determine which class a word type is associated with in the context of a single topic, such that the sign signifies association with the positive or negative class and the magnitude signifies the strength of association. Notice that typical feature selection metrics, such as χ^2, rank word types based on their discriminative power between

3 The Amazon Product Reviews corpus contains 24 categories which produce 276 unique category pairs.

(a) 2500-2500 / 2500-2500

Category	Class	*positive*	*negative*
	Books	2500	2500
	Pet Supplies	2500	2500

(b) 500-500 / 4500-4500

Category	Class	*positive*	*negative*
	Movies and TV	500	500
	Pet Supplies	4500	4500

(c) 500-4500 / 4500-500

Category	Class	*positive*	*negative*
	Movies and TV	500	4500
	Pet Supplies	4500	500

(d) 4500-500 / 4500-500

Category	Class	*positive*	*negative*
	Movies and TV	4500	500
	Pet Supplies	4500	500

(e) 500-1000 / 8000-500

Category	Class	*positive*	*negative*
	Movies and TV	500	1000
	Pet Supplies	8000	500

(f) 8000-1000 / 250-750

Category	Class	*positive*	*negative*
	Movies and TV	8000	1000
	Pet Supplies	250	750

Table 3: Different splits of data sampled from the Amazon Product Reviews dataset. The splits are created in such a way that each split tests specific aspects in the ensemble. Split (3a) is a baseline condition where both the categories and the classes are balanced. In (3b) the classes are balanced but the categories are imbalanced. In (3c) the categories and classes are balanced overall but the class distribution is flipped between the two categories. In (3d) categories are balanced but the classes have a large imbalance. Splits (3e) and (3f) should best reflect real world data sets where both the categories and the classes are imbalanced.

two classes, but the metrics do not tell *which* class each word type is associated with. Knowing the class association is crucial for our purposes.

The agreement measure is based on PMI which for two words w_i and w_j is defined as $\log \frac{p(w_i,w_j)}{p(w_i)p(w_j)}$, where $p(w_i, w_j)$ is the joint probability of seeing words w_i and w_j together and $p(w_i)$ and $p(w_j)$ are the marginal probabilities of seeing the i^{th} (j^{th}) word. PMI measures the independence of two random variables: if two words w_i and w_j occur independently of each other their PMI value will be 0; if the words only occur together – they are perfectly correlated – then $p(w_i) = p(w_j) = p(w_i, w_j)$ and PMI is positive. In general, PMI is positive if the words occur together more than one would expect by chance $p(w_i, p_j) > p(w_i)p(w_j)$ and negative otherwise.

We can change PMI such that it measures the association of a word with a class label by setting $\text{PMI}(w_i, y^+) = \log \frac{p(w_i,y^+)}{p(w_i)p(y^+)}$, where the marginal probability of the positive

sentiment class ($p(y^+) = p(y = 1)$) can be measured on the word level as the ratio between the number of word occurrences in documents of positive sentiment and the number of word occurrences overall[4]. This allows us to identify words that have a strong positive or negative correlation with a class.

As noted before, in order to determine which class a word type is associated with we would like to have a metric whose sign signifies the direction of association and magnitude the strength of association. We can achieve this by measuring the difference in correlations between the positive and negative classes: $\text{PMI}(w_i, y^+) - \text{PMI}(w_i, y^-)$. We will denote this as $\delta_{\text{PMI}}(w_i, y^\pm)$. The value is close to 0 for words that rarely co-occur with either class – infrequent words – and for words that are strongly correlated with both classes, for instance, stop words. Large positive values indicate a strong correlation with the positive class and large negative values indicate a strong correlation with the negative class.

Adding an indicator for a corpus allows us to identify which data set the words come from. Let $\delta^{t_0}_{\text{PMI}}(w_i, y^\pm)$ denote the agreement of word w_i from topic t_0. We can now measure the agreement of words W between two topics t_0 and t_1 as

$$\mathit{agreement}(t_0, t_1, W) = \frac{1}{|W|} \sum_{i=1}^{|W|} \delta^{t_0}_{\text{PMI}}(w_i, y^\pm) \delta^{t_1}_{\text{PMI}}(w_i, y^\pm) \tag{26}$$

For individual word types that do not differ in their class association between the topics $\mathit{agreement}(t_0, t_1, w_i)$ will be positive and for those that do it will be negative. Overall, the lower the agreement value the more word types there are that differ in their class association across topical contexts.

A potential issue in using PMI as the base for the vocabulary agreement measure is its sensitivity to word frequencies. Rewriting the formula for PMI we get $\log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} = \log \frac{p(w_i|w_j)}{p(w_i)}$. The usual concern is that changes in word frequencies will have an undesired effect on the resulting PMI scores. For two words that are perfectly correlated $p(w_i|w_j) = p(w_j|w_i) = 1$ and $\text{PMI}(w_i, w_j) = \log \frac{1}{p(w_i)}$, words that frequent and have a high probability are down weighted in the PMI calculation. The issue could be addressed by using a normalised PMI measure (Bouma, 2009) or by accounting for term frequencies

4 $p(y^+)$ can also be measured on the document level using only document counts instead of word counts. All probabilities were estimated on the word level.

in some other manner. However, we did not find this to be an issue in practice and have therefore not explored alternative methods for computing the agreement measure.

SECTION 3.3
Evaluation

Before describing the results from our experiments we describe the evaluation metrics we use. As there are a number of different data splits with varying degrees of class imbalance, measures such as accuracy are undesirable as they makes comparisons across different data splits difficult. Accuracy does not account for class imbalance, and can give good results for one-class predictors so long as that one class is the majority class in the data set. As this is the case for a number of data sets we use a metric that remains coherent when there are changes in class balance: Matthew's Correlation Coefficient (MCC). We also use Precision, Recall and F1-score in a number of scenarios to compare the models.

MCC (Matthews, 1975) is a metric used in binary classification scenarios that measure the agreement of predicted labels with true labels. The metric is defined as

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{27}$$

where TP is true positives, FP is false positives, TN is true negatives and FN is false negatives.

MCC is related to the χ^2-statistic and can also be defined as $|\text{MCC}| = \sqrt{\chi^2/\text{N}}$ where N is the number documents. MCC takes values in the interval $[-1, 1]$, with zero being a completely random predictor, 1 being perfect prediction and -1 representing complete disagreement between predicted and true labels. The benefit of MCC over metrics like Precision or Recall is that it accounts for predictions made on both the positive and negative classes. It is therefore well suited for situations where class imbalances exist, such as those presented later in this Chapter.

Table 4 shows these evaluation metrics for a single SVM trained on 80% of the data for one category pair and all the different data splits. It is clear that Accuracy does not properly reflect the model's performance as a very substantial change in Recall, from

0.39 (4500-500 / 4500-500) to 0.91 (500-4500 / 4500-500), still yields an accuracy of 0.91 for both cases. In the latter data set there are 5000 documents in both classes, whereas in the former there is a large class imbalance with 9000 positive documents against 1000 negative documents. In the former case the classifier is over predicting the positive class, but this unwanted behaviour is not captured by Accuracy.

Similarly, we do not use Area Under the ROC Curve (AUC) as this metric has been shown to be incoherent due to its link with predicted class probabilities. The area under the Receiver Operating Characteristic (a graph showing false positive rate against the true positive rate) (ROC) curve is dependent on the tradeoff a particular model makes between the true positive rate and the false positive rate (Hanley and McNeil, 1982). The tradeoff is a reflection of the relative misclassification costs *the classifier* has determined from the training data set. Clearly, the relative misclassification costs between the classes are a feature of the problem itself, not something a classifier should learn from data. More problematically, AUC is using different measures to compare models with each other (Hand and Anagnostopoulos, 2013; Hand, 2009).

Finally, we note the large variation in performance across different topical splits of the data (Table 4). The single SVM performs best when the topical split is aligned with the class split (data set (c)) and worst when a minority class distributes over the two categories (data set (d)). We will return to this issue in Section 3.6.1.

Movies and TV - Pet Supplies						
ID	Sample Size	MCC	Accuracy	Precision	Recall	F1
(a)	2500-2500 / 2500-2500	0.64	0.82	0.81	0.83	0.82
(b)	500-500 / 4500-4500	0.63	0.81	0.82	0.81	0.81
(c)	500-4500 / 4500-500	0.82	0.91	0.91	0.91	0.91
(d)	4500-500 / 4500-500	0.44	0.91	0.62	0.38	0.47
(e)	500-1000 / 8000-500	0.66	0.92	0.80	0.62	0.70
(f)	8000-1000 / 250-750	0.61	0.89	0.74	0.61	0.67

Table 4: Five different evaluation metrics for the single SVM classifier on different data splits for the *Movies and TV vs. Pet Supplies* category pair. Each sample contains 10000 documents in total with varying class and category imbalances. Accuracy is a bad evaluation metric as substantial changes in Precision and Recall are not reflected in Accuracy as the class imbalance of the dataset changes.

SECTION 3.4

Experimental Methodology

Before presenting our empirical work with the topical ensemble we describe the experimental methodology used.

DATA PREPROCESSING The initial six data splits (Table 3) are created by sampling uniformly at random per category label from the Amazon corpus. Each of the splits is sampled independently of any other. After sampling the documents for both categories in each split, the documents are tokenised and lemmatised[5] filtering out email addresses, url links, punctuation and stopwords (Stone, Dennis, and Kwantes, 2010). Finally, we restrict the vocabulary to contain only those items that occur in less than 75% of the documents.

TRAINING / TEST DATA SPLITS For each of the six data splits we take 50 stratified random samples from the complete data set to create training, test and evaluation sets. We set the random seed to 983475. The samples are split 80/20 into training/test

5 Tokenisation and lemmatisation is performed using the english language models in spaCy. For a detailed list of the software versions used please see Appendix F.

and hold-out evaluation data. All hyper-parameter optimisation is performed on the training/test data using 10-fold cross validation (90/10 training/test split). All reported performance metrics are measured on the hold-out evaluation data. The training data splits are used to train the ensemble models only, the topic models are trained on a separate data sample of similar size sampled using the same procedure, but ensuring that the document identities differ between the ensemble training and evaluation data and the topic model's training data.

TOPIC MODEL TRAINING For each of the six data sets we train one topic model on a sample of 10000 documents. We trained a separate topic model from each product category pair ensuring that the training data for LDA is separate from the training or evaluation data of the ensemble classifiers. This is to ensure that the topic model does not inadvertently leak any information about the ensemble's evaluation data beyond that of the intended topical information. The single topic model for each data set is cached and used for all train/test samples during the ensemble training.

ENSEMBLE TRAINING For the ensemble models we test a number of different settings. Each parameter setting is trained and evaluated 25 times and the average performance metrics on hold-out data are reported. We compare the topical ensemble(s) to a strong baseline model and to an oracle that has access to the true product category assignments of the reviews. We test two variants of the topical ensemble: an unweighted majority voting ensemble and a weighted majority voting ensemble. The SVM penalty parameter C was optimised separately for each model and each training split in the ensemble using 10-fold cross validation. We also include a sanity check 1-topic case where the topic model outputs a unit vector of topic weights for all documents. Since the per instance training weights are all set to a unit vector the results should be identical to those of the single SVM baseline. The number of topics was varied between 2 (the true number of product categories) and 40.

ENSEMBLE PREDICTIONS Once the classifier ensemble has been trained there are at least two ways of producing predictions with it for unseen data: unweighted and weighted majority voting. Unweighted majority voting, or simply majority voting, is

a common ensemble prediction method. This method involves each classifier giving a binary prediction for the class label of a test document and the final ensemble prediction being the class that received the most votes. Weighted majority voting uses the topic model weights for the test data when aggregating the ensemble predictions. Each classifier's binary prediction is weighted by the document-topic probability of the corresponding topic for that document.

SECTION 3.5

Experiments - Balanced Categories and Classes

The previous Sections detailed the building blocks of a topical ensemble, how we have used those building blocks to create the topical ensembles, the data sets we use for experimentation and the metrics we use to compare the models. This Section and the ones following present experimental results analysing the performance of the topical ensemble.

We focus the analysis first on a single sub-sampled data set where the classes and product categories are balanced (Table 3a). The data set contains equal amounts of data for each of the two product categories and an equal amount of data for each class within those categories. We start with this simplified data set to establish a performance baseline for the task overall, and to flesh out details of how the ensemble predictions are formed (Subsections 3.5.1.1, 3.5.1.2 and 3.5.1.3) as well as issues regarding the training of individual models in the ensemble, specifically how the scale of the training weights impacts performance (Subsection 3.5.2).

3.5.1 *Baseline Performance*

In this Subsection we analyse the performance of the topical ensemble against the baseline and the Oracle model. Overall, the results in Table 5 show that the framework works as expected: the 1-topic sanity check case equals the performance of the single SVM. The unweighted majority voting ensemble suffers when there are very few topics, whereas the weighted majority voting ensemble suffers when there are too many topics. The last column in Table 5 shows the oracle model which performs unexpectedly badly. The following three sub sections expand on these observations.

	SVM	LDA+SVM		
Vote Aggregation		0/1	θ	†
Topics				
1	0.675	0.675	0.675	
2		0.655*	0.675	0.624**
3		0.662*	0.670	
4		0.658*	0.666*	
5		0.665*	0.666*	
6		0.664*	0.662*	
8		0.667*	0.662*	
10		0.667*	0.659*	
20		0.667*	0.657*	
30		0.665*	0.652*	
40		0.664*	0.652*	

Table 5: Matthews Correlation Coefficient of the topical ensemble against a single SVM and an oracle. Results marked with * are significantly different at the 5% level (McNemar's test, p-value < 0.05). The topic model variants are based on the way in which the ensemble votes are aggregated: 0/1 uses a simple unweighted majority and θ uses a majority vote weighted based on the document topic proportions of test documents. The last column (†) is an oracle model that has access to the gold-standard category information. The 1 topic case is a sanity check to make sure the software implementation works correctly. There is no difference between the models in the 1 topic case.

3.5.1.1 *Unweighted Majority Voting*

The unweighted majority voting ensemble is conceptually closest to a traditional bagging based ensemble model. The ensemble is trained using the topic model derived weights, but those weights are not used at prediction time. Instead, the binary predictions from the models in the ensemble are combined and the majority class vote is taken as the ensemble's final prediction.

The unweighted majority voting ensemble under performs the single SVM by between $1-2$%-points depending on topic model size. Very small ensembles (2-4 topics) perform worse than larger 10 to 20 topic ones, although the larger ensembles still clearly under perform the single model.

This raises a question regarding the source of errors: one possibility is that the ensemble predictions are tied between the two classes and the final prediction becomes a random choice. Alternatively, since the topic weights are not used during prediction the topical expertise of each model is not accounted for. "Topical experts" in the ensemble

could therefore be out-voted by models that do not understand the topical composition of a test document. We will return to these issues later in Section 3.5.1.4.

3.5.1.2 *Weighted Majority Voting*

The weighted majority voting ensemble uses the topic model derived instance weights also during prediction not just during training. The binary class predictions of each topical classifier are turned into scalar votes using the document-topic proportion of a test document as the vote magnitude. The votes for each class are then aggregated and the class with the highest vote weight becomes the final prediction.

The topic weighted majority voting matches the performance of the single model for small topic models, but the performance degrades as the number of topics increases. This is due to the document topic proportions thinning out as the number of topics increase: the training data weights become lower on average (see Figure 6) and the classifiers are learning from ever fewer data points (see Section 3.5.2). Similarly the document topic weights for the test data are spread out across more topics for larger topic models, and are on average lower for a randomly selected topic resulting in an increased number of tied predictions as can be seen in Figure 7 (cyan dotted line).

3.5.1.3 *Oracle*

Finally, we note the relatively poor performance of the oracle model, the final column in Table 5. This model has access to the gold standard product category assignments for each document and the oracle performance should give an indication of how well the ensemble can be expected to perform if the underlying topic model did a perfect job at separating the two product categories. However, the oracle is much worse than either of the ensemble versions.

This discrepancy is due to the training data weights. In our instantiation of the ensemble training, the weights are used to inform the learning algorithm how much attention to pay to each learning instance. In the case of the oracle these weights are either 1 or 0 for in-category and out-of-category documents respectively. In other words, the oracle does not pay any attention to errors committed on the out-of-category data. For the balanced dataset (Table 3a) this means that each of the two classifiers in the ensemble learn from only that half of the training data which corresponds to its product category. Ide-

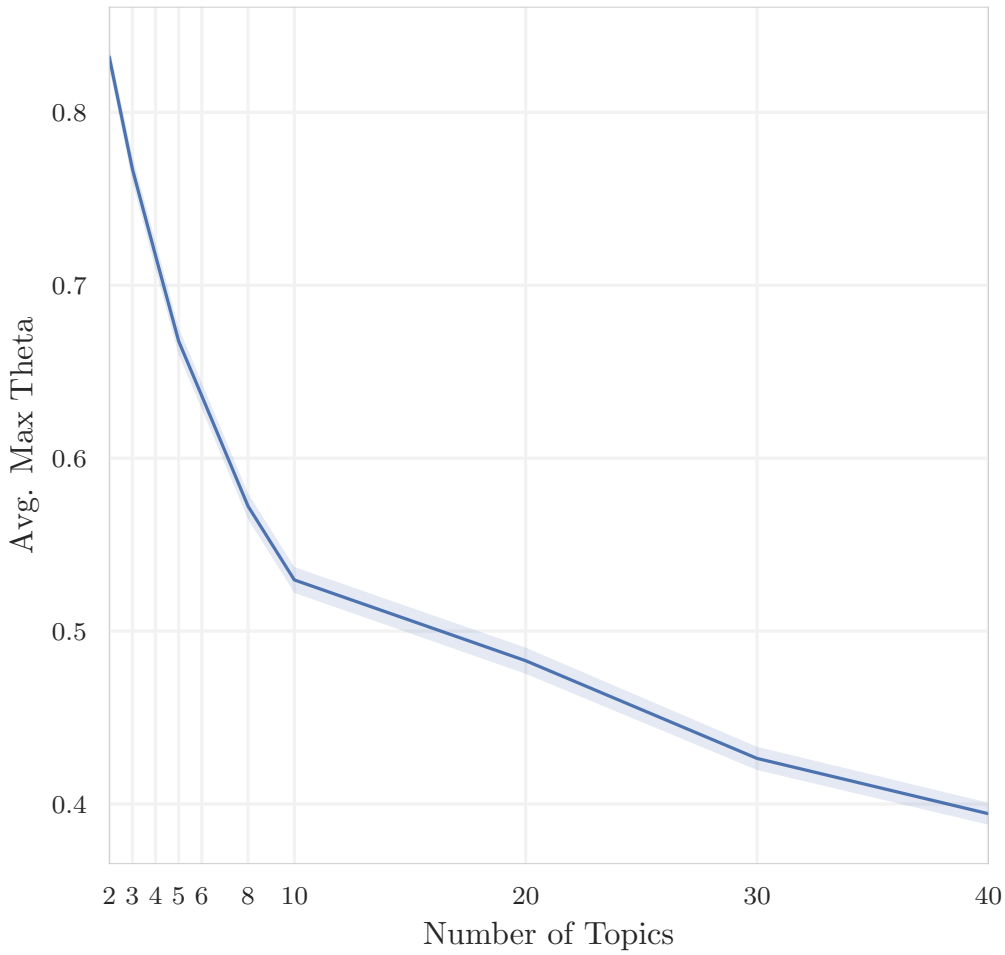


Figure 6: Average maximum document topic weight across the ensemble with the 95% confidence interval. When the max weight drops below 0.5 a single "expert" model can be out-voted by the rest of the ensemble when using weighted majority voting.

ally, the model would not ignore errors on out-of-category data but instead simply pay more attention to errors on in-category data. This can be achieved by changing the scale of the learning weights from the range $[0, 1]$ to a range that starts from 1 (see Section 3.5.2).

3.5.1.4 *Tied Predictions and Errors Committed by the Ensembles*

We return now to the issue of where the unweighted and weighted ensembles commit errors and why. Recall that the unweighted ensemble commits fewer errors as the size of the ensemble grows but the weighted ensemble has the opposite behaviour. This is

clear from Figure 7 which also shows that the number of errors grows faster than the number of tied predictions. Tied predictions are ones where the class votes split evenly between the two classes. We defined an even split for the unweighted ensemble to be cases where the vote difference is less than two, and for the weighted ensemble cases where the difference is less than 0.15. This allows us to treat ensembles with an even or odd number of models equally. We set the threshold empirically such that if less than two thirds of the overall ensemble vote weight is on one class the prediction is tied.

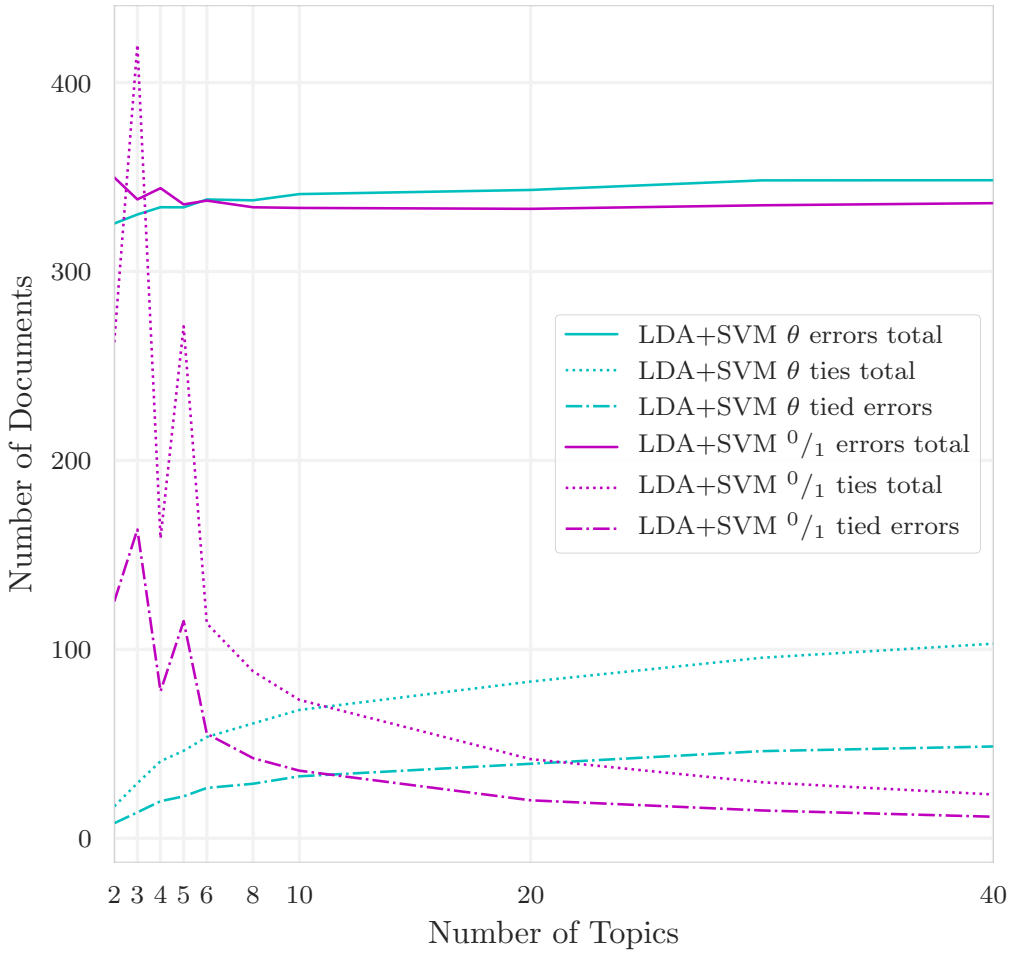


Figure 7: Average number of errors committed by the two ensemble model variants using unscaled LDA document-topic proportions. The number of tied predictions and the number of errors due to tied predictions are displayed as dotted and dash-dotted lines respectively, the unweighted ensemble is shown in magenta and the weighted ensemble in cyan. Total size of the test set is 2000 documents, giving an error rate of approximately 17% on the balanced dataset (Table 3a).

For the weighted ensemble the number of errors grows faster than the number of tied errors suggesting that the weighted voting is working correctly: models contribute to the final class prediction proportional to their expertise about the topic of a test document. However, the weighted ensemble commits more errors on all ensemble sizes above 8 compared to the unweighted ensemble.

Resolving tied predictions might offer a good way of improving the performance of both ensembles, but as the proportion of errors out of all tied predictions is approximately 50%, it is not clear that resolving the tied predictions would improve the ensemble's performance as the correct tied predictions would also get resolved, likely introducing new errors.

3.5.2 *Scale of Learning Weights*

The previous Section showed that the topical ensemble works as intended, but the oracle performance highlighted an issue in the scale of the learning weights, a crucial aspect in building the ensemble. As the instance weight is part of the objective function (Equation 23) of the classifier, very low weights overall will cause the learning algorithm to simply minimise the norm of the coefficient vector. Since the output of the topic model for each document is a sparse topical encoding with most values close to 0, using the topic model output as is can lead to under fitting the models.

For a large topic model most of the topic proportions are typically close to zero, which, if given to the learning algorithm as is, will cause the algorithm to heavily discount errors on those documents in the training set and focus on minimising $||w||^2$, the norm of the coefficient vector. The result of this can be seen in Table 5. Both the unweighted and weighted ensembles under-perform the baseline single SVM. Furthermore, the oracle ensemble is significantly worse than all three other models. The oracle has exactly two classifiers, matching the true number of product categories. Each classifier in the oracle ensemble has seen all of the training data, but, due to the binary weights, in optimising the objective function the models have accounted for errors committed on only half the data.

This problem is exacerbated by increasing the number of topics as the topic distributions become ever thinner. A topically biased SVM will on average have seen less data

during training than a single SVM trained with the whole unweighted corpus. The problem can be addressed by adjusting the dynamic range of the learning instance weights. The absolute scale of the weights does not matter as optimising the overall C parameter will account for any uniform changes, but the relative scale between all the instance weights does matter. The scaling can be a simple linear transformation of the weights into a new range.

In this Subsection we present results from experiments where we changed the scale of the learning weights. We tested a number of different scales for in-topic and out-out-topic documents, including $[0,1]$, $[0,2]$, $[1,2]$ and $[1,4]$. A scale of $[0,1]$ is a baseline condition where the document topic probabilities are scaled between 0 and 1, i.e. not changed. A scale of $[0,2]$ slightly modifies the baseline condition to make errors on documents that have a high document topic probability twice as costly without changing the cost of errors on out-of-topic data. In the cases where the lower end of the scale is 1 or more the classifiers will learn from the same amount of the data as the baseline model, but will weight errors on in-topic data higher than those committed on out-of-topic documents.

Note that although the dynamic range of $[1,2]$ is the same as that of $[0,1]$ the two scales are not the same. Specifically, they differ in how errors on out-of-topic documents are treated: the former places the same importance on errors committed on out-of-topic documents as a vanilla SVM would, whereas the latter weight scaling causes the topical SVM to down weight errors on out-of-topic documents relative to the baseline SVM classifier. The results are summarised in Table 6 (the full set of results is displayed in Table 23 in the Appendix).

Overall, we find that the range of the training weights has a significant impact ($p < 0.05$, McNemar's test) on the ensemble's performance, especially when the size of the ensemble grows (Table 6). The negative impact of increasing the topic model size for the weighted majority voting model is mitigated and in some cases reversed by changing the scale of the training weights. Note that this experiment did not change the weights used during prediction; those are still the unscaled document-topic weights from the topic model.

It is noteworthy that the performance of the weighted 20 topic model jumps from 0.667 to 0.680 ($p < 0.05$, McNemar's test) simply by changing the scale of the training

		SVM	LDA+SVM		
Vote Aggregation			0/1	θ	†
Topics	Weight Scaling				
1		0.675	0.675	0.675	
2	[0, 1]		0.655	0.675	0.624
	[0, 2]		0.654	0.674	0.622
	[1, 4]		0.671*	0.678	0.666
	[2, 3]		0.675*	0.676	0.675
5	[0, 1]		0.665	0.666	
	[0, 2]		0.664	0.664	
	[1, 4]		0.676*	0.677*	
	[2, 3]		0.678*	0.676*	
10	[0, 1]		0.667	0.659	
	[0, 2]		0.666	0.660	
	[1, 4]		0.679*	0.678*	
	[2, 3]		0.678*	0.677*	
20	[0, 1]		0.667	0.657	
	[0, 2]		0.666	0.656	
	[1, 4]		0.679*	0.680*	
	[2, 3]		0.678*	0.677*	

Table 6: Matthew's Correlation Coefficient for the balanced data set ((a) 2500-2500 / 2500-2500). The weight scale settings that are significantly better than the [0/1] weight scaling for the corresponding model settings are marked with a superscript * ($p < 0.05$, McNemar's test.)

weights from raw topic model output to $[1, 4]$. In the latter case each of the 20 models is using all the training data available but is biasing the training data to documents closely related to the topic by a factor of 4.

The benefit of scaling the learning weights is clear from the performance improvement of the ensemble models compared to their non-weighted (weight scale [0/1]) counterparts. There is however only a small improvement in Matthew's Correlation Coefficient (MCC) from 0.675 to 0.680 for the 20 topic model with training weight scale of $[1, 4]$ compared to the single SVM. This improvement is not statistically significant at the 5% level ($p = 0.14$, McNemar's test).

3.5.2.1 *Error Analysis*

An error comparison between the ensemble and the single SVM shows that the 2 models have a slight difference in their respective error profiles. Although most of the errors are committed on the same documents, 11% to 12% of the errors are unique to either

model (Table 7). This suggests that further performance improvements could be made by combining the predictions of the single SVM with those of the ensemble. In the best case scenario those 11% of errors could be reduced such that only the common errors remain (column 00 in Table 7), but identifying how to combine the predictions of the models is a challenge.

Topics	Weight Scale	00	01	10	11
2	[0, 1]	261.8	63.3	63.6	1611.3
	[1, 4]	275.1	49.9	47.1	1627.8
4	[0, 1]	252.3	72.7	81.7	1593.2
	[1, 4]	276.1	49.0	47.0	1627.9
10	[0, 1]	248.9	76.2	92.1	1582.9
	[1, 4]	281.9	43.1	40.4	1634.5
20	[0, 1]	244.9	80.1	98.2	1576.7
	[1, 4]	284.9	40.1	35.4	1639.6

Table 7: Average agreement in numbers of documents for predictions between SVM and weighted ensemble. The columns are: 00 number of documents where both models made an error, 01 number of documents where only SVM made an error, 10 number of documents where only the ensemble made an error and 11 number of documents where both models made the correct prediction.

An option would be to look at the errors committed by the ensemble due to it being uncertain, i.e. cases where the votes split roughly evenly between the target classes. In the case where the ensemble prediction is tied there are at least three valid ways of breaking it. The ties can be solved by using the single SVM as another predictor in the ensemble, although this may introduce new ties. The prediction of the model with the highest document-topic probability to the test document can be used, or the ties can be broken by using the single SVM predictions in cases where the ensemble cannot make a decision. We tested all of the above tie braking methods but found no improvement in performance beyond what the weight scaling already provides.

Looking at the errors committed by the weighted 20 topic ensemble (Figure 8) - so far the best performing model - we see that the errors are split unevenly between cases where the ensemble prediction is tied and those where it is not, with only a small fraction (approximately 4%) of errors committed on a tied vote. Problematically, the tied predictions are split evenly between erroneous ones and correct ones: any change to the

tied predictions will therefore correct some errors, but also introduce new ones. It is not clear how the tied votes could be resolved to improve performance.

Overall, across all the different parameter combinations, we found a small but not significant ($p > 0.05$, McNemar's test) improvement in performance compared to the single SVM baseline on the balanced dataset using 80% of the data for training (Table 6, 20 topics, weight scale $[1, 4]$).

3.5.3 *Summary*

In this Section we compared the performance of the topical ensemble to a single SVM on a data set that does not contain any category or class imbalances. We showed that the unweighted ensemble model performs better with a larger number of topics, whereas the weighted ensemble performs better with a small number of topics.

Changing the scale of the instance weights used during ensemble training has a significant impact on model performance. The unscaled document-topic proportions are ill suited for training weights as they cause the SVM to ignore or at least discount errors committed on potentially substantial parts of the training set. Changing the weight scale such that errors on all training documents are taken into account, but that errors for in-topic documents are more costly, allows the ensemble models to match the performance of the single SVM.

The errors committed by the single SVM baseline model and the ensemble are unique in approximately 11% of error cases, suggesting that combining predictions from the two could be beneficial. However, it is not clear how the predictions should be combined as simply adding the predictions of the single SVM to those of the ensemble is likely to introduce new errors while fixing some old ones.

We turn our attention next to the imbalanced datasets to see how category or class imbalances change the performance characteristics of the ensemble in comparison to the baseline method.

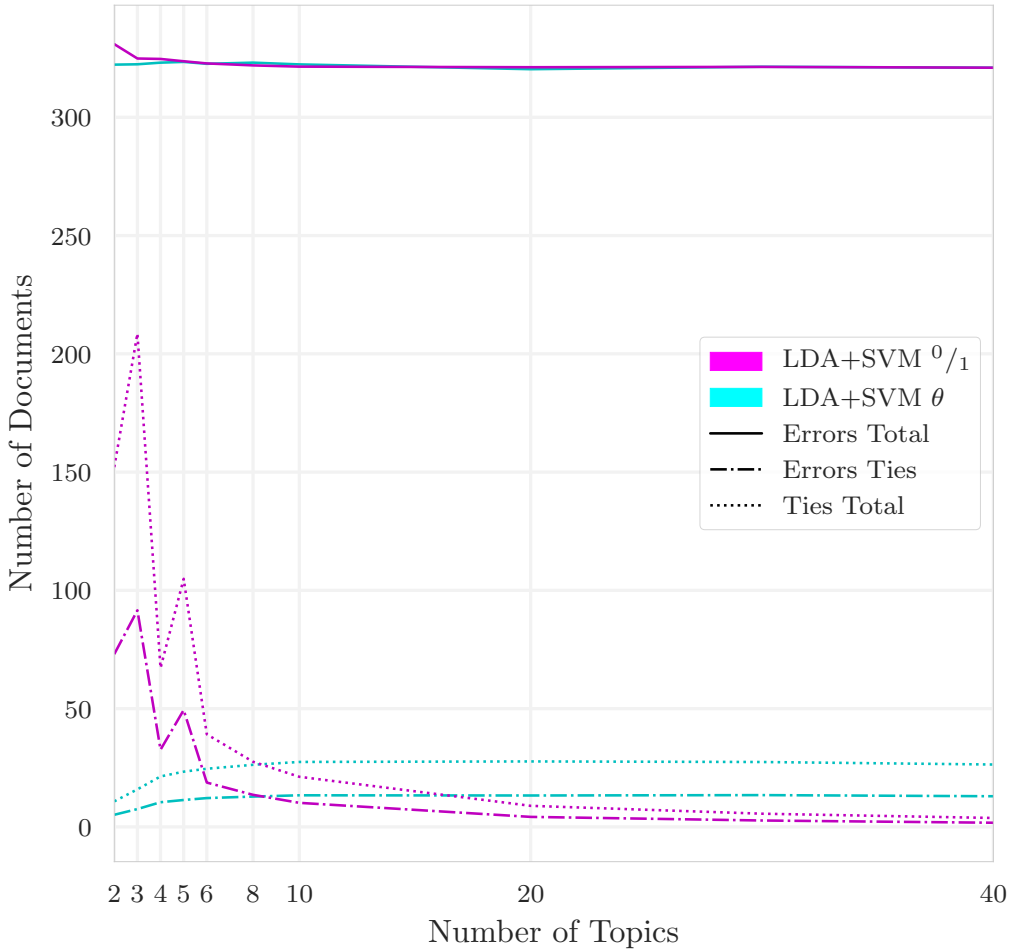


Figure 8: Average number of errors committed by the two ensemble model variants with weight scaling [1, 4]. The number of tied predictions and the number of errors due to tied predictions are displayed as dotted and dash-dotted lines respectively. A tied prediction for the unweighted ensemble is one where the class votes are less than 2 votes apart, and for the weighted ensemble less than 0.15 apart.

SECTION 3.6

Experiments - Category and Class Imbalance

The previous Section compared the performance of the topical ensemble to a single SVM on a data set with no topical or class bias: both classes and both product categories had the same number of documents. Class and category imbalances, however, clearly have an impact on the performance of the single SVM. Tables 4 and 8 show that the single SVM performance is highly variable depending on how the target classes distribute across the product categories. The best performance (0.812 MCC) is achieved on a dataset where the class balance is flipped between the two categories ((c) 500-4500 / 4500-500); on this dataset predicting a class label is aligned with predicting a product category, i.e. words that are highly topical and correlate with reviews of only one of the two product categories also correlate with only one of the two classes. This observation supports the notion that a misalignment between the class distribution and topical categories hampers the performance of a classifier and that correcting for that misalignment can be beneficial.

In this Section we explore the issue of class and category imbalances in more detail. The experimental framework is the same as in the previous Section, but the experimentation is expanded to data sets where the categories or classes or both are no longer balanced. To highlight the differences in the data sets we first analyse the performance of the single SVM (Section 3.6.1), and then in Section 3.6.2 we compare the topical ensemble to the single SVM.

3.6.1 *Single SVM Performance*

The relationship between the target classes and topical categories can have an impact on classifier performance. The experiments described in Section 3.5 were performed on an artificially balanced data set. Here we will look at how category and class imbalances, and more specifically their interaction, impact classifier performance. In total, we created six different data sets to investigate how the interactions between the categories and classes impact performance. These data sets are shown in Table 3.

The target classes are balanced overall in each of the three datasets (b), (c) and (d); data sets (e) and (f) contain a mixture of data that is more reflective of real world scenarios, i.e. there is a minority class label that distributes unevenly across the category structure. Table 8 shows the performance results of the single SVM on the unbalanced data sets.

ID	Sample Size	MCC	Accuracy	Precision	Recall	F1
(a)	2500-2500 / 2500-2500	0.675	0.837	0.834	0.843	0.838
(b)	500-500 / 4500-4500	0.682	0.841	0.839	0.844	0.841
(c)	500-4500 / 4500-500	0.812	0.906	0.907	0.905	0.906
(d)	4500-500 / 4500-500	0.482	0.919	0.648	0.425	0.510
(e)	500-1000 / 8000-500	0.671	0.921	0.794	0.644	0.710
(f)	8000-1000 / 250-750	0.628	0.900	0.775	0.606	0.679

Table 8: Single SVM performance on different data splits over all 8 category pairs. Each sample contains 10000 documents in total with varying class and category imbalances.

3.6.1.1 *Data set (b) 500-500 / 4500-4500*

In data set (b) the target classes are balanced in the data set overall as well as within each category. The categories, however, are imbalanced by a 1/9 ratio. The category imbalance means that if a model represents data only in the larger category it will suffer at most a 10% penalty on performance by misclassifying all of the data belonging to the smaller category. The results in Table 8 show that the single SVM performs slightly better on this data set than on the balanced one (data set (a)), both in terms of MCC and Precision, Recall and F1-score. Having most of the data come from a single topical area seems to be beneficial.

3.6.1.2 *Data set (c) 500-4500 / 4500-500*

Data set (c) contains an equal amount of data for both categories and for both classes, but the class distribution between the categories is flipped: one category is mostly positive data while the other is mostly negative data. Modelling the classes is therefore well

aligned with modelling the categories. On this data set the single SVM has the highest performance.

3.6.1.3 *Data set (d) 4500-500 / 4500-500*

Data set (d) contains a minority class that distributes evenly over the categories, with both categories being balanced. Here 90% of the data is concentrated on one class, with roughly the same class distribution within each category. As both classes distribute over both categories, modelling only one of the categories will result in poor performance. Indeed we see that the single SVM performs worst out of all data sets on this one. Notice that this data set is the complement to data set (b), but instead of one category being the majority, one class is the majority.

3.6.1.4 *Data sets (e) and (f)*

Data sets (e) and (f) reflect the kind of imbalances one would expect to find in real world data sets: imbalanced classes distributed over an uneven category structure. The single SVM performance is roughly in line with those achieved on data sets (a) and (b) although the models have focussed more on the negative class. This can be seen in MCC being in line with those of (a) and (b) while Precision and Recall are lower. MCC accounts for True Negatives (TN) while Precision and Recall only focus on the positive class.

3.6.2 *Topical Ensemble Performance (b) - (d)*

We will now compare the topical ensemble to the single SVM on data sets (b) through (d). The results are shown in Tables 9 and 11.

Overall, we find that the 20 topic ensemble with a weight scale of $[1, 4]$ is consistently the best performing ensemble. The difference to the baseline single SVM model is, in most cases, small and not statistically significant. The data set where the target classes are imbalanced ((c) 4500-500 / 4500-500) is the exception to this rule. On this data set the difference is statistically significant ($p = 0.0125$, McNemar's test). Curiously the topic weighted majority voting (θ) ensemble that takes the document topic weights into account also during prediction does not perform as well as the unweighted ma-

jority voting method, suggesting that the document topic weights are perhaps not as informative as one might have expected.

		SVM	LDA+SVM	
Vote Aggregation			0/1	θ
Sample Size Topics	Weight Scaling	MCC		
(b) 500 - 500 / 4500 - 4500				
2	[1, 4]	0.682	0.681	0.681
	[1, 10]	0.682	0.682	0.682
20	[1, 4]	0.682	0.685	0.683
	[1, 10]	0.682	0.684	0.683
40	[1, 4]	0.682	0.685	0.684
	[1, 10]	0.682	0.685	0.683
(c) 4500 - 500 / 4500 - 500				
2	[1, 4]	0.482	0.480	0.481
	[1, 10]	0.482	0.480	0.479
20	[1, 4]	0.482	0.491	0.486
	[1, 10]	0.482	0.491	0.486
40	[1, 4]	0.482	0.491	0.484
	[1, 10]	0.482	0.490	0.485
(d) 500 - 4500 / 4500 - 500				
2	[1, 4]	0.812	0.811	0.812
	[1, 10]	0.812	0.811	0.813
20	[1, 4]	0.812	0.814	0.814
	[1, 10]	0.812	0.814	0.813
40	[1, 4]	0.812	0.814	0.814
	[1, 10]	0.812	0.814	0.814

Table 9: Summary Table of Matthews Correlation Coefficient comparing the ensemble model to the single SVM for datasets (b), (c) and (d).

3.6.2.1 *Topical Bias or Random Variation?*

The question arises whether the improvement of the unweighted majority voting ensemble is due to the topic weights used during training or possibly an artifact of random variations in the training of the SVMs themselves. To investigate the impact of the document topic weights as training weights we trained a dummy ensemble that is otherwise identical to the topical ensemble with the exception of unit document topic weights during training. The unit weights remove any potential information derived from the topic model while still creating an ensemble of classifiers. The results in Table 10 show that

the topical ensemble models have no statistically significant difference to the control model (LDA+SVM†) and indicate that the performance improvement achieved by the ensemble is potentially due to random variations in the SVM training as opposed to the topical bias of the classifiers (Table 10). This finding is in line with existing literature on ensemble models and offers an alternative explanation to the findings of Xiang and Zhou (2014). They showed an approximately 2%-point improvement using a similar topical ensemble, but did not test if the improvement was due to the topical bias or random variation.

		SVM	LDA+SVM		LDA+SVM†
Vote Aggregation			$0/1$	θ	
Sample Size Topics	Weight Scaling		MCC		
(a) 2500 - 2500 / 2500 - 2500					
2	[1,4]	0.675	0.671	0.678	0.672
20	[1,4]	0.675	0.679	0.680	0.676
(b) 500 - 500 / 4500 - 4500					
2	[1,4]	0.682	0.681	0.681	0.683
20	[1,4]	0.682	0.685	0.683	0.683
(c) 4500 - 500 / 4500 - 500					
2	[1,4]	0.482	0.480	0.481	0.481
20	[1,4]	0.482	0.491	0.486	0.490
(d) 500 - 4500 / 4500 - 500					
2	[1,4]	0.812	0.811	0.812	0.811
20	[1,4]	0.812	0.814	0.814	0.811

Table 10: Summary Table of Matthews Correlation Coefficient comparing the single SVM and our ensemble model to an ensemble where each SVM is trained with unit weights for all training data (LDA+SVM†). Note that the weight scaling does not apply to the LDA+SVM† as all the weights are set to one for that model. The differences are not statistically significant at the 5%-level.

The training method used in (Xiang and Zhou, 2014) sub-samples the training data based on document topic weights instead of using the weights directly in the error function of the classifier. To ensure that the alternative training methodology was not causing the observed differences, we trained topical ensembles using the sub-sampling method with a sampling threshold of 0.01. We found that the ensembles trained using the sub-sampling method performed worse across all topic model sizes. Additionally we found that as the number of topics increased the performance of the sub-sampled

ensemble would reduce, regardless of voting mechanism used. This makes sense as the document topic probabilities are spread out over an increasing number of topics and fewer documents exceed the threshold of 0.01 for any specific topic. Each classifier in the ensemble therefore gets less training data as the size of the ensemble increases. The results are available in Table 24 in the Appendix.

3.6.3 *Topical Ensemble Performance (e) - (f)*

Finally for the last two data sets ((e), (f)) that reflect the kinds of class and category imbalances observed in real world data, i.e. situations where both the classes and categories are imbalanced overall in the entire corpus, we see the same patterns as before (Table 11). The unweighted majority voting ensemble yields a slight improvement over the single SVM. The improvement on data set (e) is statistically significant at the 5% level ($p = 0.0439$, McNemar's test). As before, the same performance improvement is observed when training the ensemble with unit weight vectors instead of the document topic weights.

3.6.4 *Summary*

In this Section we looked at how class and category imbalances impact the ensemble and single SVM performance. We found that class and category imbalances have a significant impact on the performance of the single SVM and that in some cases the topical ensemble does improve the performance. However, contrary to previous research (Xiang and Zhou, 2014) we found that the improvement is likely due to random variations in the ensemble training as opposed to the ensemble utilising the topical information. This suggests that the individual SVMs in the ensemble are not able to use the topical side information about training instances to improve performance over the baseline single SVM. This does not mean that the topical information is not useful in general, but that perhaps the classifier performance has reached a plateau. The next Section looks at the learning curve of the ensemble.

		SVM	LDA+SVM	
Vote Aggregation			0/1	θ
Sample Size Topics	Weight Scaling			
(e) 500 - 1000 / 8000 - 500				
2	[1, 4]	0.671	0.655	0.669
	[1, 10]	0.671	0.649	0.665
20	[1, 4]	0.671	0.676	0.674
	[1, 10]	0.671	0.676	0.673
40	[1, 4]	0.671	0.676	0.675
	[1, 10]	0.671	0.676	0.676
(f) 8000 - 1000 / 250 - 750				
2	[1, 4]	0.628	0.614	0.627
	[1, 10]	0.628	0.606	0.623
20	[1, 4]	0.628	0.634	0.632
	[1, 10]	0.628	0.634	0.630
40	[1, 4]	0.628	0.635	0.632
	[1, 10]	0.628	0.634	0.631

Table 11: Summary Table of Matthews Correlation Coefficient comparing the ensemble model to the single SVM across a number of different datasets.

SECTION 3.7

Learning Curve

The previous Sections focused exclusively on a setting where 80% of the data was used for training, with the remaining 20% kept as an evaluation set. One crucial aspect for machine learning models is the amount of training data needed to reach a certain level of performance. As all the training data needs to be labelled, usually by human annotators, creating training data for new models can be costly and slow. Therefore, in this Section we look at how the performance of the ensemble degrades compared to the single SVM as the amount of training data is gradually reduced while keeping the evaluation data fixed.

We ran experiments varying the amount of training data from 5% to 80% with 10% increments between 10% and 80%. Each sub-sample was taken randomly using stratified sampling to ensure the class distribution of the samples was the same as for the whole data set. All evaluation was done on the same 20% sample of data. The topic models

were trained on a separate sample of the corpus, exactly as before, and only the training data for the SVM ensemble was changed. The results are shown in Figures 9 and 10.

Overall, we find that the performance improvement of the single SVM classifier flattens out when approximately 60% (6000 documents) of the data set is used as training data, with the exception of the (c) data set where the performance saturation happens at about 70% training data (Figure 9). The topical ensemble improves over the single SVM performance across the range up to about 60% training data with larger improvements for the smaller amounts of training data (Figure 10 shows relative improvement over the baseline).

On all datasets the topical ensemble has a statistically significant improvement over the baseline at the 5% significance level (McNemar's test), up to about 50% training data and on some data sets more than that. The largest and most consistent performance improvements are achieved on datasets (d), (e), and (f).

The unweighted majority voting performs better than the weighted majority voting on almost all data sets and training data sizes. The few exceptions are cases where the two models perform equivalently.

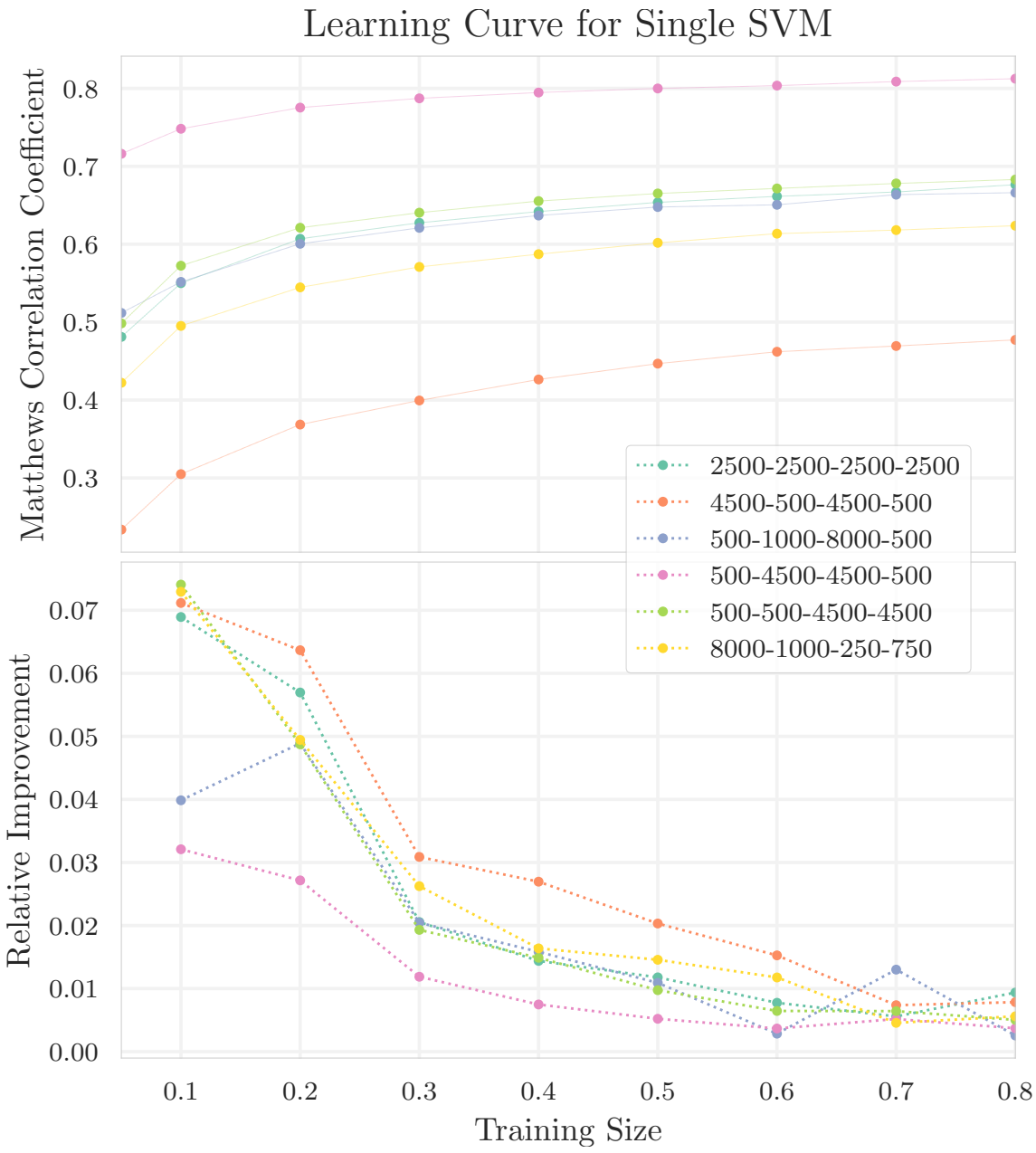


Figure 9: Absolute (top) and relative (bottom) improvements in single SVM performance, measured as Matthews Correlation Coefficient, on all of the Amazon Product Review datasets.

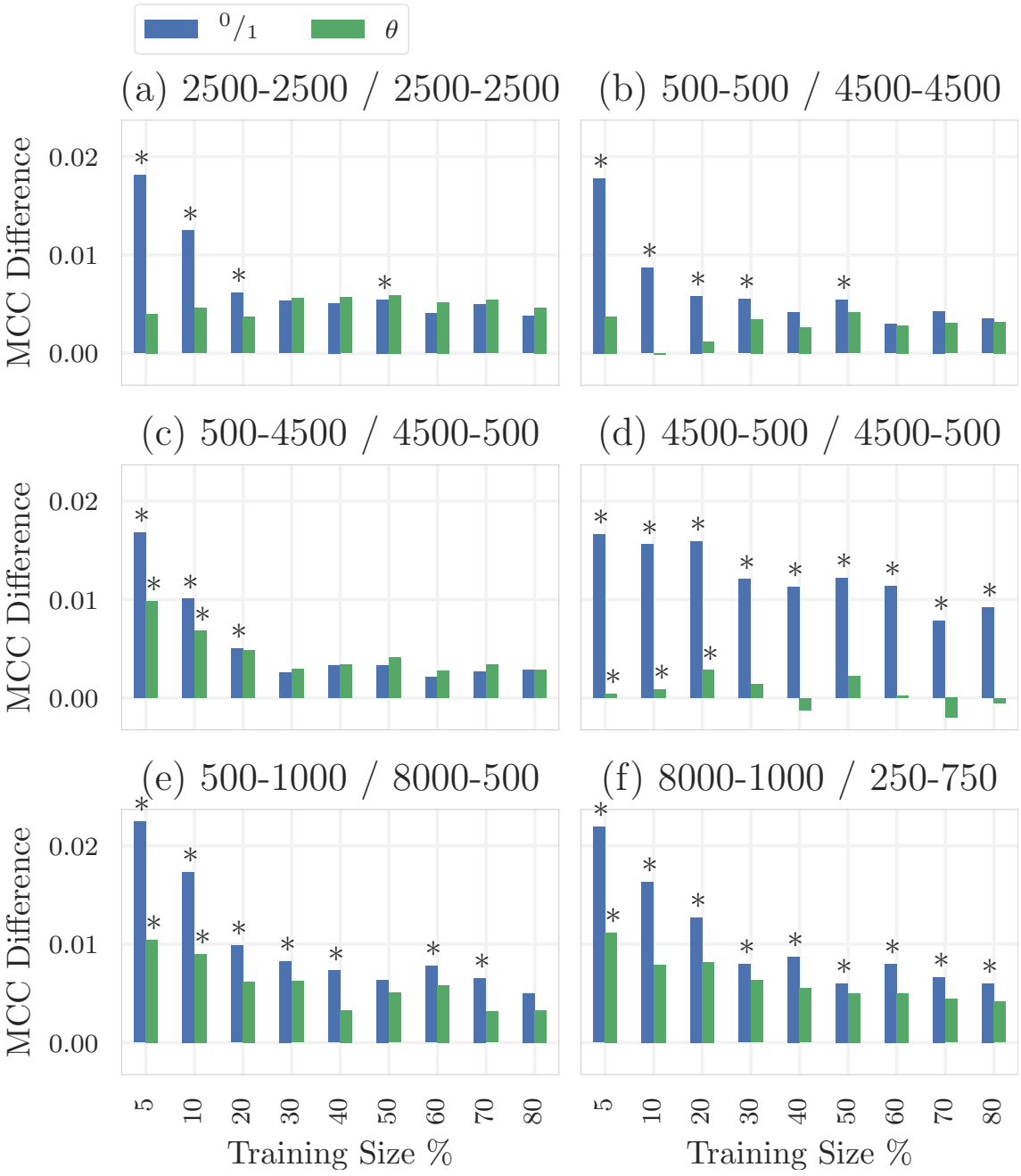


Figure 10: Relative improvement of the ensemble (20 topics, weight scale [1, 4]) compared to single SVM. Blue is unweighted majority voting, green is document-topic weighted majority voting. The results marked with a * are statistically significant at the 5% level (McNemar's test).

SECTION 3.8

Summary

We presented and tested an ensemble model that is based on the topical context of a corpus. The motivation is to utilise the topical context as a non-local document feature and allow the individual classifiers in the ensemble to become topical experts. Crucially, the classifiers are not trained to identify the topic of the documents but rather to classify the document into some other classes such as positive or negative sentiment.

We showed that in a topically diverse corpus imbalances in the class and/or topic distribution of the documents have a large impact on the performance of the classifier. The better the alignment is between the target classes and the topical categories the better the performance of the classifier tends to be and vice versa. We also showed that an inconsistency in the classification signal between topical contexts has a negative impact on the performance of a classifier.

Our hypothesis was that accounting for the topical structure in cases where the class structure of a corpus is not aligned with the topical structure should improve classifier performance. This hypothesis is supported by previous research (Xiang and Zhou, 2014). Although we did observe a statistically significant improvement over the baseline when using an ensemble, on closer inspection the topical information is likely not the source of the improvement. This finding is interesting as it contradicts the previously published results and suggests that the improvement shown by Xiang and Zhou (2014) is not necessarily due to the topical information.

For the topical information to help the classifier there would need to be a high correlation between important document features and topic association. The correlation would in turn translate into the separate topical SVMs using topic specific features. We haven't observed the topical differences translating into coefficient vectors that are themselves topically skewed, suggesting that although there exist inconsistencies in the class association of features across topical contexts, those inconsistencies are not sufficiently resolved by knowing the topical context. The local SVMs end up not having a lot of differentiation in the high / low coefficient features.

4

TOPICAL ENSEMBLES FOR HIERARCHICAL MULTI-LABEL CLASSIFICATION

In this Chapter we focus on the application scenario of assigning documents to possibly overlapping topical categories. This task is applicable, for instance, to a service where users can annotate news articles with tags that the users themselves find meaningful, applying as many tags as they see fit to each document. The users create a taxonomy of labels which can then be used by a machine learning system to annotate new articles and give recommendations about content likely to be of interest to the users.

Another example is online services that offer researchers the possibility of annotating published academic papers with arbitrary keywords that are outside those used by the original publisher[1]. Similarly to the news articles, a machine learning model can learn from the labelled data and predict which new articles belong to the annotated categories, thus organising new and unseen articles under known category labels. These services allow creating ad-hoc dynamic label hierarchies that help users of the services organise content in a way that is unique to each user. We present an ensemble model that can learn the label hierarchies and apply them effectively in a real world setting.

The labels describe topics such as Politics, Sport or Tennis and form a hierarchy. For instance, Tennis and Football would both be subclasses of Sport. In general, the scenario is a hierarchical multi-label classification problem, a problem that common classification algorithms can not, without modifications, deal with. Our ensemble is competitive with existing multi-label algorithms and has some desirable features that traditional algorithms do not. Specifically, our model is able to leverage large quantities of unlabelled data to learn a broad topical representation of documents; this allows the ensemble to

1 Services such as CiteULike (http://citeulike.org/), Mendeley (http://mendeley.com) and ResearchGate (https://www.researchgate.com/) allow adding user defined tags to articles.

learn from fewer labelled examples than comparable algorithms, and model labels that are topically highly specialised. A large corpus of unlabelled data and specialised topical labels are both key aspects of the problem setting addressed in this Chapter.

The application scenario highlights a number of constraints that need to be addressed. The label set for any corpus in this scenario should be considered transient: since the labels are user-defined and relate to real world events, changes to the label set are likely. Consider a label such as Science & Environment applied to news articles. Articles likely to be annotated with that label would cover a broad range of issues from theoretical physics to climate change. Now consider an oil disaster on the Gulf of Mexico: before, articles talking about a specific oil company would likely not be related to the Science & Environment label, but after the accident that likelihood would change and this change should be reflected in the model. Similarly, new labels, that before had no meaning, are created all the time. Examples include Gamergate[2], #MeToo and many others. The phenomenon is not specific to news. Academia experiences similar bursts of interest in specific methodologies: a label such as Deep Learning for instance had very little meaning before 2008.

It is important for a real world system to be able to pick up on these kinds of changes quickly and reliably. It would, therefore, be beneficial if adding new labels did not require relearning the entire model as this can be costly and may introduce changes to the model's performance on parts of the label hierarchy that have already been learned. The transient labels also mean that a perpetual cold-start problem exists. Since any new label will, by definition, have a limited amount of training data, the classification model should be able to learn new labels from as few labelled instances as possible. Additionally, due to the perpetual cold-start problem, it would be beneficial if new labels for which little training data exists could "borrow" information from similar already learned labels. These constraints serve as guidelines for our experiments and the analysis of results.

The rest of the Chapter is structured as follows. First, in Section 4.1 we describe the general problem of multi-label learning and the specific problem addressed in this Chapter, we then describe our approach to the problem (Section 4.2). Section 4.3 introduces the data set we use for experiments and Section 4.4 describes a number of experiments

2 https://en.wikipedia.org/wiki/Gamergate_controversy

we conducted and their results. Finally Section 4.5 concludes the Chapter with a Summary of the experimental results.

SECTION 4.1

Multi-label Learning

Many real world applications involve a set of labels that are not mutually exclusive and, in some cases, form a hierarchy; examples include ecological habitat modelling (Levatić, Kocev, and Džeroski, 2015), functional genomics (Alaydie, Reddy, and Fotouhi, 2012) and document categorisation (Rubin et al., 2012; Li, Ouyang, and Zhou, 2015; Levatić, Kocev, and Džeroski, 2015). The research community has developed new algorithms and modified existing ones to deal with the multi-label problem. The multi-label classification problem is a generalisation of single-label classification to one where each instance is annotated with one or more labels from a finite set. Unlike in binary or multi-class classification the labels for a document are not mutually exclusive.

We follow the description of Madjarov et al. (2012) for multi-label classification. An algorithm that solves a single-label classification task needs to learn a mapping from examples $\mathbf{x} \in \mathcal{X}$ (where $\mathcal{X}$ is the space of all possible examples) to a label $\lambda_j \in \mathcal{L}$ for each instance, where $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots \lambda_q\}$ is a set of mutually exclusive labels. A training set E_{tr} consists of pairs of items $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{L})$ where y_i is a single label. When the total number of labels q, is two, the problem is a binary classification problem. When $q > 2$ it is a multi-class problem. In the multi-label scenario each document $\mathbf{x}$ is mapped to a subset of labels $\mathcal{Y} \subseteq \mathcal{L}$ where the labels in $\mathcal{L}$ are not mutually exclusive. The training set E_{tr} consists of pairs of items $(\mathbf{x}_i, \mathbf{y}_i)$ where $\mathbf{y}_i$ is a binary vector of length q, with all label indices corresponding to the labels assigned to an instance marked with a 1 and everything else with a 0. For any document $\mathbf{x} \in \mathcal{X}$ the relevant or correct labels are $\mathcal{Y}$ and $\mathcal{L} \setminus \mathcal{Y}$ is the set of irrelevant labels.

Multi-label learning involves two closely related problems: learning to rank and learning to classify. Approaches to multi-label learning often rely on ranking all the available labels per instance, and then using some thresholding mechanism – either heuristic or learned – to select a set of predicted labels. In some cases an out-of-order subset of labels is selected.

Tsoumakas and Katakis (2007) divide the multi-label learning approaches into ones that modify an existing algorithm and ones that transform the output label space and create several binary classification sub-tasks that can be dealt with using common algorithms. The adapted algorithms usually solve the classification problem by ranking all the labels for each instance based on a scoring function $f : \mathcal{X} \times \mathcal{L} \rightarrow \mathbb{R}^{\mathcal{L}}$, and selecting the top N elements from the ranked list of labels or using a threshold value to determine the appropriate labels. Other approaches aim to directly select a label set that satisfies a metric, for instance the maximum a posteriori probability of labels of the k-nearest neighbours (Zhang and Zhou, 2007).

Label transformation methods create a number of binary classification tasks by modifying the label sets of documents. The label sets are modified such that a number of binary classification tasks are created from the multi-label data. The resulting binary tasks can be dealt with using traditional binary classifiers organised in an ensemble. Common label transformation approaches include the OvR ensemble and the One versus One (OvO) ensemble. In OvR the binary classification tasks are created using each label in $\mathcal{L}$ in turn as the target label and all the other labels as the "other" class. However, because the documents can be annotated with more than one label the "other" class should ideally be constrained to the set of labels that did not co-occur with the target label. An OvR ensemble has q classifiers, one for each of the labels in $\mathcal{L}$. In an OvO ensemble the binary classifiers are created from pairs of labels. This keeps the label semantics clear, but creates $O(q^2)$ binary tasks and corresponding classifiers. Additional approaches are reviewed in Section 2.6.

In Table 12 we list the precision @k for a number of recent state-of-art models. Unfortunately these metrics are not comparable with ours as our model does not produce a ranking of labels, but instead produces discrete label assignments. Furthermore, the precision @k metric can be misleading as it often favours large categories that are high in the category hierarchy. As our aim is specifically to investigate how topical information impacts the classification performance of categories at each level of a label hierarchy we can not use precision @k as an evalution metric.

	FastXML	FastText	BoW-CNN	XML-CNN
Precision @1	0.95	0.95	0.96	0.97
Precision @3	0.78	0.80	0.81	0.81
Precision @5	0.55	0.56	0.57	0.56

Table 12: Precision @K metrics for state-of-the art model in multi-label classification on the RCV1 dataset. As our model only creates discrete label assignments, not a ranking, we are not able to measure the precision @k metric.

4.1.1 *Evaluation Methods*

The multi-label learning scenario is challenging also from the point of view of evaluation. As the labels are not mutually exclusive, common evaluation metrics such as Accuracy, F1-score, Precision and Recall have been redefined as instance based metrics for the multi-label case. Additional metrics such as Zero-One-Loss ($0/1$-loss) and Label Ranking Loss (Ranking Loss) also exist. All of these measures are used in the existing literature with little consistency between different scenarios or research projects. The number of different evaluation metrics used reflects the difficulty of evaluating multi-label learning algorithms. Similarly to single label scenarios, a labelling can be incorrect because of a missing label or an extra label. However, in contrast to single label problems, there can be multiple missing or extra labels for any instance. This increases the number of ways in which a labelling may be incorrect and introduces the possibility of partially correct labellings. Furthermore, some multi-label problems have closely related labels for which the label relations may need to be accounted for. It has been suggested that using a single metric is not sufficient and that multiple metrics should be used for any given problem-algorithm pair (Madjarov et al., 2012). In this work we use $0/1$-loss, per category (binary) precision and recall as well as their multi-label counterparts (defined below).

We picked these metrics as they allow us to analyse the performance of a classifier with respect to the constraints of the application scenario (see Section 4 for a discussion). $0/1$-loss and multi-label precision and recall are useful for evaluating the performance as a whole. However, we are also interested in knowing how the classifiers perform on each of the tiers in the label hierarchy or each individual category. We use the binary per category precision and recall metrics for the latter purpose.

Given true label assignments for $\mathbf{y}_i \in \{0,1\}^q$ that encode the correct labels for each document $\mathbf{x}_i$ as binary vectors of size q and a predictor $z(\mathbf{x}_i) : \mathcal{X} \to \{0,1\}^q$ that outputs label predictions, the evaluation metrics can be defined as follows

$$\text{precision} = \frac{1}{N} \sum_{i=1}^{n} \frac{|y_i \wedge z(\mathbf{x}_i)|}{|y_i \vee z(\mathbf{x}_i)|} \tag{28}$$

$$\text{recall} = \frac{1}{N} \sum_{i=1}^{n} \frac{|y_i \wedge z_i|}{|y_i \vee (\neg y_i \wedge z_i)|} \tag{29}$$

where $|y_i \vee (\neg y_i \wedge z_i)|$ is the number of false negatives, $|y_i \vee z_i|$ is the number of predictions or equivalently true positives plus false positives, and $|y_i \wedge z_i|$ is the number of true positives. Note that in the multi-label case both $|\mathbf{y}_i|$ and $|z_i|$ may be greater than one, whereas in the binary case both are exactly one. $0/1$-loss is defined as

$$L_{0/1} = \frac{1}{N} \sum_{i=1}^{n} I(y_i = z_i) \tag{30}$$

where $I(\cdot)$ is the indicator function.

A further complication in evaluating multi-label classification algorithms is the relatedness of the labels themselves. Label correlations are problem specific but are an important feature of many problems. Document categorisation is one example, as misclassification costs between different pairs of labels vary. From the point of view of a user, misclassification of related categories such as Sports and Football carries a lesser penalty than misclassification of unrelated categories like Sports and Politics or Football and Politics. In hierarchical document categorisation, the directionality of the misclassification is also important: classifying a (Football) document as only (Sports) is a mistake only in the sense of not being specific enough, whereas misclassifying a document with the labels (Sports, Tennis) as (Football) is clearly an error. Applying the parent label (Sports) only would have been better. It is not clear in the existing literature how these kinds of issues could be factored into the evaluation of multi-label methods. We note the

difficulties in evaluating hierarchical document classification scenarios for completeness. We do not develop new evaluation methods in the work presented in this Chapter.

SECTION 4.2

Topic Based Multi-label Classifier

So far we have described the problem of multi-label classification and outlined common evaluation metrics for multi-label classification as well as detailed the specific application scenario we address. We identified three key issues arising from the application scenario: new labels that an already existing model needs to subsume, the cold start problem associated with new labels and correlations between labels in the label set. In this Section we describe our solution and discuss how it solves each of these problems. The fundamental building block of our solution is an unsupervised topic model (Latent Dirichlet Allocation)[3]. For the purposes of this discussion we will refer to the output of the topic model as *topic*s while *tag*s are the set of pre-defined target categories of which a subset is assigned to each document in the labelled data set.

Our approach is motivated by the problem scenario and shortcomings in existing multi-label approaches: multi-label ensemble methods (OvR, OvO) do not adjust well to situations where the label hierarchy grows over time. For an OvR ensemble the definition of the "other" class becomes increasingly complex as new "other" labels are added and an OvO ensemble suffers from an exponential growth in computational cost. General purpose multi-label algorithms, such as Multi-label K Nearest Neighbours (ML-KNN) or Decision Trees do not account for label correlations or take advantage of features of our problem scenario. Specifically, the algorithms do not account for the topical use of language. In general, our approach utilises a continuous distributed document representation created by an unsupervised topic model coupled with an effective weight learning paradigm and a decision function that takes advantage of the label hierarchy.

Since the topic model is trained on unlabelled documents a large corpus of historical data can be gathered. The historical data allows the model to capture the coarse topical structure of documents. The coarse topical structure is likely to remain stable over the short to medium term, allowing updates to the topic model to be infrequent. Furthermore, the distributed document representation can be used to distinguish fine

3 For a review of LDA please refer to Section 2.5.3.

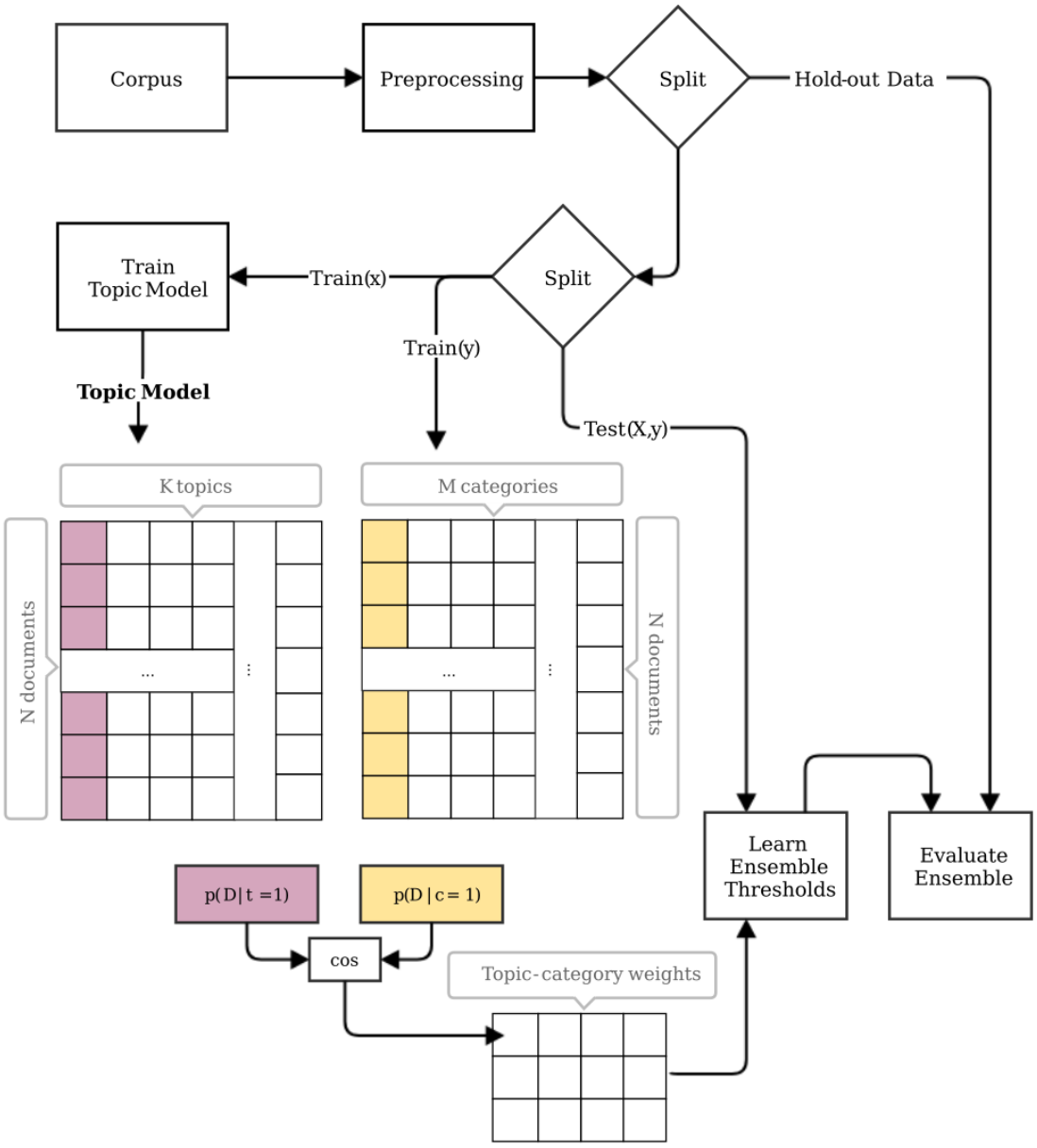


Figure 11: Training workflow for a topical ensemble. Unlabelled training data is used to train a topic model. Labelled training data is used to compute similarities between topic weights and category assignments. The category assignments are normally discrete meaning that $p(D|c = 1)$ is a vector of binary values, but the method can handle probabilistic category assignments as well.

grained differences between target categories by using weighted combinations of topics. Any single topic is unlikely to match a specific tag exactly. We therefore compute a set of association weights for each tag. These weights are computed using a vector similarity metric and a small amount of labelled data. A classifier for a single tag consists of the weights and a decision function that can optionally utilise information from the label hierarchy. The model as a whole consists of many such classifiers, one for each tag, organised in an ensemble.

We will now discuss in detail how the model is trained. Recall that for every document the topic model output is a probability distribution over topics $\mathbf{d}_i \in \mathbb{R}^K$ (the document vector). The matrix $D \in \mathbb{R}^{K \times |E_{tr}|}$ contains the document probabilities of all documents in the training set E_{tr}. The vector $\mathbf{t}_j \in \mathbb{R}^{|E_{tr}|}$ for a single topic contains the document-topic probabilities of the labelled documents. This vector is a description of which documents in the labelled data set are closely related to topic j; we call this the *topic vector*. The topic vector is a topic specific, unnormalised score distribution over all documents. It encodes which documents are closely related to, or "belong" to, a given topic. Finally, the *category vector* $\mathbf{c}_m \in \{0,1\}^{|E_{tr}|}$ contains the gold standard label assignments for each category and describes which documents in the labelled data have been labelled with tag m. Note that the topic and category vectors have the same dimensionality. We compute the association weights between each tag and all topics using a similarity metric between the *topic vector* and the *category vector*.

Using any pair $(\mathbf{t}, \mathbf{c})$ of topic and category vectors we compute the similarity between the topic and category vectors as

$$\text{sim}(\mathbf{t}, \mathbf{c}) = \text{cosine similarity}(\mathbf{t}, \mathbf{c}) = \frac{\sum_{i=1}^{N} g(\mathbf{t}_i)\mathbf{c}_i}{\sqrt{\sum_{i=1}^{N} g(\mathbf{t}_i)^2}\sqrt{\sum_{i=1}^{N} \mathbf{c}_i^2}} \tag{31}$$

where $g(\cdot)$ is a threshold function that sets values less than 0.0015 to 0. The threshold function is applied to the topic vector, because in practice the topic model tends to produce somewhere between 2-5 meaningful topic probabilities for any document. The other values are in the order of 0.0015 or below. Semantically these low values are equal to 0, i.e. the document does not "belong" to the topic. However, as the values are arith-

metically not 0 they complicate the similarity calculation. Without the threshold, cosine similarity tends to produce a large number of noisy topic-tag weights due to a large number of documents in the labelled data set that do not belong to a tag *and* have a low document-topic probability for any given topic.

The weight matrix from the similarity computation forms a $K \times Q$ matrix W, where K is the number of topics and Q is the size of the label set. The K weights for each individual tag are separate from the weights of any other tag, allowing tags to be added or removed without impacting the performance of the model on already learned labels. The weights between K topics and a single tag together with a decision function form a single label classifier. The model as a whole consists of Q such single label predictors.

Finally, unseen documents are labelled by first ranking all tags and then applying a decision function to the ranked tags. Given the weight matrix W and the document-topic probabilities for an unseen document, a score is assigned for each tag j by applying the similarity function to weights $\boldsymbol{w}_j$ and the document-topic distribution of a test document. The scores are ranked and a decision function is applied to the ranked tags to select which tags to apply. Since the tags form a hierarchy, we compare decision functions that optionally utilise the hierarchy. Applying a label to a document requires learning a label threshold. The thresholds are tag specific and are learned using an unseen test set and a loss function such as $0/1$-loss.

SECTION 4.3
Data sets

We used the Reuters Corpus version 1 (RCV1) (Lewis et al., 2004) for all our experiments. The corpus consists of approximately 800000 news articles collected from the Reuters news service between August 1996 and August 1997. Each article has been manually annotated with topic categories, industry categories as well as geographical regions. The topic and industry categories form a hierarchy. We use the topic categories as the target labels in all experiments. The categories are summarised in Table 13. Note that the sum of the *Size* column in Table 13 is larger than the size of the data set because many of the documents belong to more than one category and are therefore counted multiple times.

Topic Code	Size	Explanation	Depth	Parent
CCAT	374316	Corporate/Industrial	1	CCAT
C11	24325	Strategy/Plans	2	CCAT
C12	11944	Legal/Judicial	2	CCAT
C151	81875	Accounts/Earnings	3	C15
C1511	23212	Annual Results	4	C151
ECAT	117539	Economics	1	ECAT
E12	27078	Monetary/Economic	2	ECAT
E121	2182	Money Supply	3	E12
GCAT	234873	Government/Social	1	GCAT
G15	19152	European Community	2	GCAT
G159	40	EC General	3	G15
GCRIM	32219	Crime, Law Enforcement	2	GCAT
GJOB	17241	Labour Issues	2	GCAT
MCAT	200190	Markets	1	MCAT
M13	52972	Money Markets	2	MCAT
M131	28185	Interbank Markets	3	M13
M132	26752	Forex Markets	3	M13

Table 13: Topic codes, category sizes and the topic code explanation for a selection of topic codes from the RCV1. The full table can be found in the Appendix (Table 25)

Each document in the dataset is annotated with a possibly overlapping set of topic category labels from Reuters topic hierarchy. The average number of labels per document is 3.19 with a maximum of 17 labels and a minimum of 0. The category sub-trees are not mutually exclusive as approximately 14% of the documents contain labels from more than a single category sub-tree. Figure 12 shows the correlations between the 1st tier labels. There is, for instance, a strong connection between the *GCAT* (Government / Social) and *ECAT* (Economics) categories: 35.9% of the documents in *ECAT* are also labelled with *GCAT*. These overlaps are important as they indicate the relatedness of categories and inform how severe errors between certain category pairs are.

Figure 13 shows the category overlaps for the 2nd tier labels. Here we see, for instance, that *GJOB* (Labour Issues) and *E41* (Employment Labour) have a strong overlap with *C42* (Labour). The overlap is unsurprising given the category explanations: GJOB (Labour Issues), E41 (Employment/Labour), C42 (Labour). These 2nd tier labels are all in different sub-trees of the overall category hierarchy.

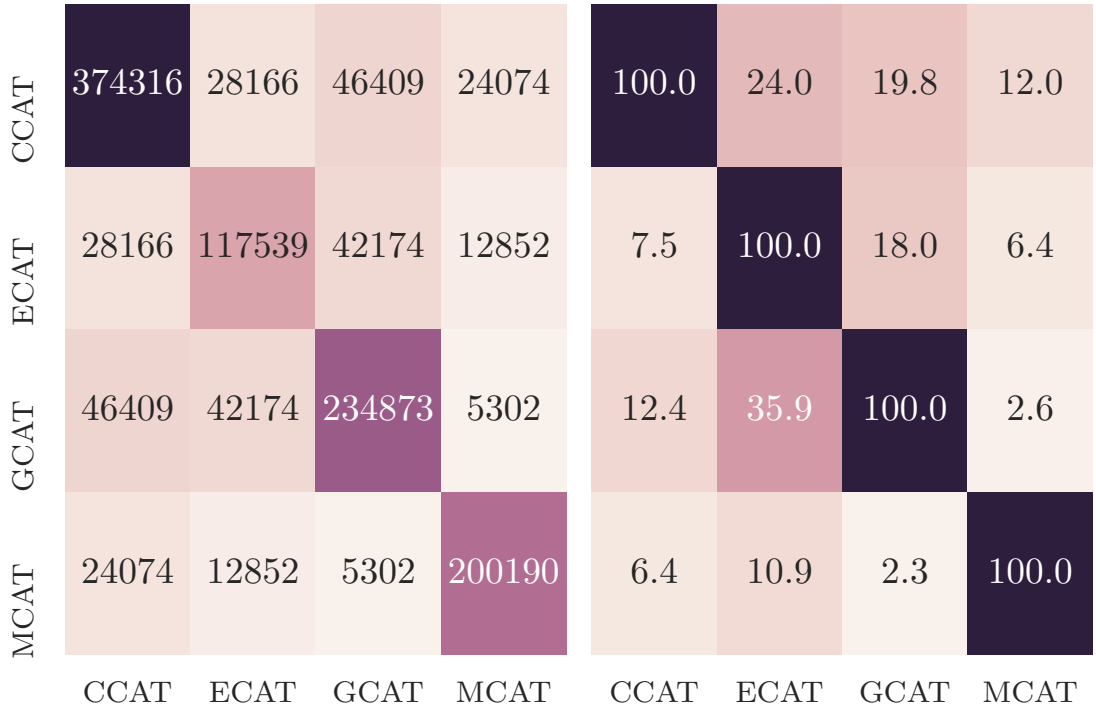


Figure 12: Category overlap for the 1st tier labels as absolute document counts (left) and percentages (right).

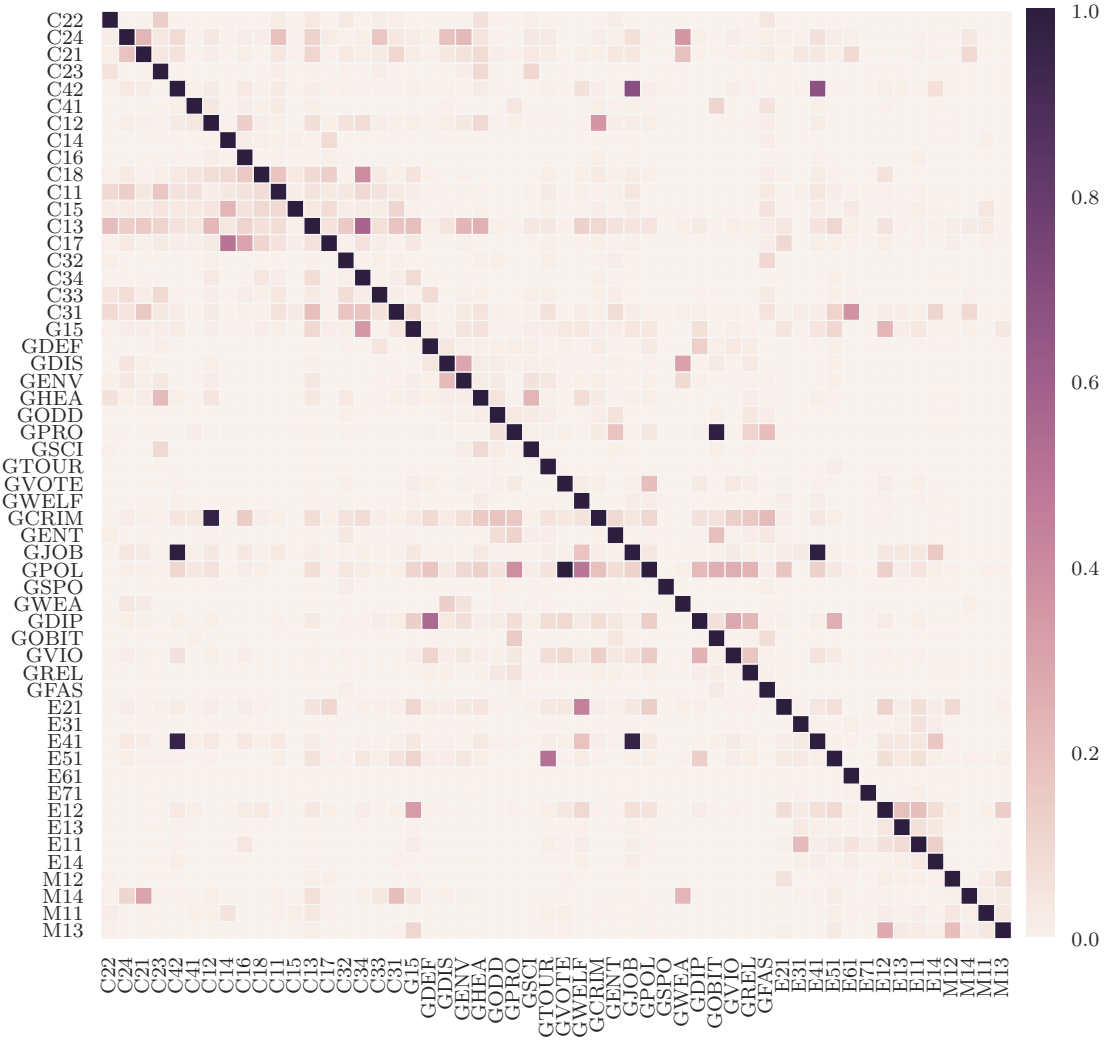


Figure 13: Percentage category overlap for the 2nd tier labels.

SECTION 4.4

Experiments

4.4.1 *Data Sets and Preprocessing*

From the set of categories listed in Table 13 we first selected the top 2 levels in the hierarchy, resulting in a dataset with 58 target labels. The *GMIL* category is ignored as it has only 5 documents. The documents in these categories were then sub-sampled to create training and test sets by taking a uniform random sample of 20 documents from each of the target categories. We then split that set of documents into training and test portions with a ratio of $1/5$, ensuring that both the training and test sets contained at least 2 documents for each of the 58 target labels. If some categories contained no documents in either, the sampling was repeated until all categories did. The sampled data is summarised in Table 14. This entire process was repeated 25 times to create 25 different data splits. We report all results are reported as averages over the 25 random samples. The same topic models were used for all data splits. The training data for the topic models was separate from the training data sampled for the experiments.

In addition to the 2-level hierarchy labels we also created a data set with target labels from the 4-level hierarchy. The data is the same as for the experiments on the 2-level hierarchy, with the exception of added labels for all documents (Table 17).

Before training the models we preprocessed the data by tokenising, lemmatising, part-of-speech tagging and performing named entity recognition[4] the documents. Each lemma is joined with its part-of-speech and named entity filtering out everything that is not marked as a noun, verb, adjective, adverb or conjunction. This feature extraction process was found to be effective for topic modelling in early empirical work. Finally, features that occur in more than 80% of the documents are ignored.

4.4.2 *Comparison Models*

We compared our model to a number of standard classification algorithms: Decision Trees, Random Forests, the Extra Trees classifier and a one-vs-rest Logistic Regression

4 Tokenisation and lemmatisation was performed using the spaCy natural language toolkit. Please see Appendix F for a detailed listing of the software components used.

classifier ensemble using a standard bag-of-words representation (LR-OvR^{bow}). As logistic regression is not directly suitable for multi-label problems, the OvR ensemble modifies the algorithm by training a binary classifier for each target label by setting the *other* class – the non-target class – to be those documents that have not been labelled with the target label. The final predictions are produced by each classifier in the ensemble predicting *True* or *False* for the inclusion of their category label for each document in the test set. Note that this is not the same as regular majority voting, as no averaging of the votes is performed. Instead, each model decides on its own whether to apply its label or not. The comparison models were all trained on a bag-of-words document representation.

Of the comparison models the Logistic Regression ensemble (LR-OvR^{bow}) performed the best, so we additionally trained a Logistic Regression ensemble on the topic model output (LR-OvR$^{\theta}$) to see how well the best performing comparison model does when given a document representation that captures document topicality. The LR-OvR$^{\theta}$ ensemble is trained on the output of the LDA model using the document-topic probability distribution for each document as the document representation instead of a bag-of-words representation. The model is motivated by research in using dimensionality reduction techniques such as LSI or LDA to train document classifiers on dense topically oriented document representations (Srinivas, Supreethi, and Prasad, 2009). The model serves as an interesting comparison as it provides insight to the benefit of training a single model on a topical document representation versus our model which trains a separate classifier for each topic. We denote the model as LR-OvR$^{\theta}$.

For our approach we trained the topic models on two different data samples: one on a random sample of 100000 documents from the target corpus and another on the entire target corpus, excluding the training data used in the experiments. For each sample we trained a topic model with 200 topics and symmetric priors. These two models are denoted as LDA 100k and LDA 800k respectively. We tested three different variants of the LDA ensemble: one where the predictions are performed on a flattened label hierarchy, and one where the label hierarchy is taken into account during prediction optionally applying mutual exclusion to the target labels at each level of the hierarchy. The hierarchical model assigns labels from sub trees only when the root of the sub tree was also predicted. The third variant imposes a mutual exclusion on the hierarchical

predictions, meaning that only a single label from any level of the tree is predicted. This takes into account the way in which the label hierarchy is used by human annotators. Approximately 86% of the gold standard labels follow this pattern. The hierarchical models are denoted with an 'H', and the mutual exclusion is signified by '*mutex*'.

The models described above were tested on two different tasks: first using only the top 2 levels from the label hierarchy and then using all 4 levels. These two scenarios are designed to highlight differences in how the models behave when new labels are added to the label hierarchy. Increasing the depth of the label hierarchy, and consequently the number of labels, complicates the task for two reasons: first the "other" class for the OvR ensemble becomes increasingly complex as the number of labels increases. Our proposed LDA based ensemble, on the other hand, should scale well as the predictors are all independent from each other. Second, as the depth of the category hierarchy increases the labels in the leaf nodes become very specialised. For instance, the G15 on the 2nd-tier is generally about the European Community, whereas G152 on the 3rd-tier is specifically about EC Corporate Policy and G157 is about EC Competition/Subsidy. We wish to know the extent to which representing the topical structure of a corpus allows modelling specialised labels such as EC Corporate or EC Competition/Subsidy. Subsections 4.4.3 and 4.4.4 explore these questions.

4.4.3 *2-level hierarchy*

Using these 25 splits we measured the 0/1-loss of all the comparison methods and our model variants. The results are summarised in Table 15.

The results show that the LDA based ensemble, in all its variants, has a lower 0/1-loss than the comparison models. Of the comparison models the Decision Tree and the OvR Logistic Regression classifier using the topic model output perform the best, but are 7 points behind in 0/1-loss compared to the best performing topic based ensemble. The 0/1-loss is an extremely unforgiving metric as all of the labels for a test document need to be correct, with no missing or additional labels. We therefore also show Ranking Loss (Tsoumakas, Katakis, and Vlahavas, 2010) which measures, as an average over all samples, the number of times incorrect labels are ranked higher than correct labels. The

Category	Training Size	Test Size	Total
CCAT	189.88	759.12	374316
C11	13.20	53.68	24325
C12	10.12	43.12	11944
C13	36.80	142.64	37410
...			
C42	14.12	57.88	11878
ECAT	143.08	572.68	117539
E61	4.36	16.44	391
GCRIM	20.68	87.24	32219
GDEF	9.64	35.92	8842
...			
GJOB	24.44	101.48	17241
GOBIT	5.56	21.16	844
GPRO	11.80	45.04	5498
...			
GWELF	5.80	21.08	1869
MCAT	58.84	241.76	200190
M11	8.20	34.92	48700
...			

Table 14: Mean number of documents per category for each category in the 2-level hierarchy over 25 random samples.

Model	0/1-loss	Precision	Recall	Ranking Loss
Decision Tree	0.92	0.45	0.37	0.54
Random Forest	0.99	0.64	0.23	0.11
Extra Trees	0.98	0.66	0.25	0.10
Logistic Regression BoW	0.91	0.67	0.51	0.15
Logistic Regression θ	0.91	0.79	0.42	0.08
LDA 100k	0.92	0.58	0.55	0.07
LDA H 100k	0.92	0.51	0.50	0.07
LDA H mutex 100k	0.80	0.70	0.48	0.07
LDA 800k	0.93	0.56	0.55	0.07
LDA H 800k	0.93	0.49	0.50	0.07
LDA H mutex 800k	0.79	0.71	0.49	0.07

Table 15: 0/1-loss, precision, recall and label ranking loss for all models on documents labelled with the top 2 levels of the Reuters label hierarchy. For the 0/1-loss lower is better. Precision and recall are measured as the average per document precision and recall, i.e. the multi-label variants of the metrics. The comparison ensemble models have 200 estimators in them, and the topic based ensembles use a topic model with 200 topics.

label ranking loss column in Table 15 shows that the LDA based ensembles are better at ranking correct labels above incorrect for test documents.

Taking the label hierarchy into account does not improve performance in label assignment, but imposing mutual exclusion between labels, which requires accounting for the hierarchy, yields the lowest 0/1-loss at 0.80 for the topic model trained on 100000 documents and 0.79 for the topic model trained on 800000 documents; the difference between the two is not statistically significant ($p = 0.1164$).

The results in Table 15 are averages over all instances and are biased towards the large categories on the 1^{st} tier. To gain a better understanding of the performance on individual labels we also analysed the models on each category separately using the common binary precision and recall metrics and averaging the results over all categories. The results are summarised in Table 16. As before, the LDA ensembles clearly outperform the comparison models. However, the hierarchical LDA model with mutual exclusion performs worse than the equivalent model without mutual exclusion. This is explained by the multi-label metrics in Table 15 being biased by large categories since the metrics in Table 16 give equal weight to each category. The mutual exclusion ensembles, similarly to the logistic regression ensembles, perform well on the 1^{st} tier labels, but poorly on the 2^{nd} tier labels for which there is less training data.

Model	Precision	Recall	F1-score	Accuracy
All Labels				
Decision Tree	0.24	0.18	0.19	0.94
Random Forest	0.20	0.03	0.04	0.95
Extra Trees	0.27	0.04	0.05	0.95
LR-OvA^{bow}	0.43	0.30	0.33	0.95
LR-OvA$^{\theta}$	0.47	0.13	0.18	0.96
LDA 100k	0.47	0.43	0.42	0.95
LDA H 100k	0.46	0.36	0.37	0.95
LDA H mutex 100k	0.53	0.29	0.33	0.96
LDA 800k	0.46	0.45	0.43	0.95
LDA H 800k	0.45	0.37	0.38	0.95
LDA H mutex 800k	0.54	0.30	0.34	0.96
Tier 1				
Decision Tree	0.62	0.54	0.57	0.75
Random Forest	0.83	0.41	0.48	0.79
Extra Trees	0.87	0.43	0.50	0.80
LR-OvA^{bow}	0.75	0.70	0.72	0.83
LR-OvA$^{\theta}$	0.85	0.68	0.73	0.85
LDA 100k	0.71	0.68	0.69	0.80
LDA H 100k	0.72	0.67	0.69	0.80
LDA H mutex 100k	0.86	0.62	0.72	0.84
LDA 800k	0.71	0.65	0.67	0.79
LDA H 800k	0.70	0.66	0.67	0.79
LDA H mutex 800k	0.86	0.63	0.72	0.84
Tier 2				
Decision Tree	0.22	0.15	0.16	0.95
Random Forest	0.16	0.01	0.01	0.96
Extra Tree	0.23	0.01	0.02	0.96
LR-OvA^{bow}	0.41	0.27	0.30	0.96
LR-OvA$^{\theta}$	0.44	0.09	0.14	0.97
LDA 100k	0.45	0.41	0.40	0.96
LDA H 100k	0.44	0.34	0.35	0.96
LDA H mutex 100k	0.51	0.26	0.30	0.97
LDA 800k	0.44	0.43	0.41	0.96
LDA H 800k	0.44	0.35	0.36	0.96
LDA H mutex 800k	0.52	0.27	0.32	0.97

Table 16: Per category average (unweighted macro) performance metrics for all labels and the different label tiers separated out.

Separating out the different label tiers we observe that the logistic regression classifiers outperform the LDA ensembles only for the four top level labels (Tier 1 in Table 16) that have plenty of labelled training data. Overall, the logistic regression ensemble requires more than 100 labelled training instances per label before its performance reaches or

exceeds that of the topical ensemble. We tested the performance of all models increasing the amount of training data from a nominal 20 documents per label to 200 documents, or the entire category if the category had fewer than 200 documents. This improved the performance of all the models making the bag-of-words logistic regression ensemble the best performing one. Notably however, that model is only marginally better than the best topical ensemble variant and the performance improvement is not statistically significant at the 5% level.

The performance improvement is also sidestepping the more important factor of annotation cost in this problem setting. While the labels at the top of the hierarchy are likely to always have enough training data, the labels further down the hierarchy likely will not. This is not only a matter of spending more effort to annotate documents. Moving down the hierarchy the labels become increasingly specialised, and simply finding documents that match those labels becomes problematic. Out of the entire 806791 documents in the RCV1 corpus collected over a period of 12 months the smallest category (GMIL - Millenium Issues) has 5 documents. Accumulating enough training data for the logistic regression model to become competitive on the smaller categories simply requires a lot of calendar time to pass.

4.4.3.1 *Summary*

We tested the topical ensemble against a number of standard multi-label classification algorithms on the top 2 levels from the label hierarchy. We found that the topical ensemble compares favourably against the other models, especially in situations where labelled training data becomes scarce and the label definitions start to become specialised. Increasing the amount of training data reduces differences in model performance, but comes at the cost of increased annotation effort and lost opportunity cost of not deploying the model while data is being annotated.

4.4.4 *4-level hierarchy*

To see how well the models behave with an increased number of labels we conducted an experiment using 4 levels from the label hierarchy. The data is the same as for the

Category	Training Size	Test Size	Total	Depth
CCAT	189.88	759.12	374316	1
C15	25.56	105.92	150164	2
C151	13.32	54.24	81875	3
C1511	6.60	25.36	23212	4
...				
ECAT	143.08	572.68	117539	1
E14	15.68	67.44	2086	2
E141	4.96	20.48	376	3
E142	4.52	17.80	200	3
...				
GCAT	170.92	680.60	234873	1
GDEF	9.64	35.92	8842	2
...				
MCAT	58.84	241.76	200190	1
M141	15.80	61.12	47708	3

Table 17: Mean number of documents per category for the 4-level hierarchy over 25 random samples. The full data table is available in the Appendix (Table 26)

experiments on the top 2 levels, with the exception of added labels for all documents (Table 17).

Overall changes are caused by the increased label count and the more specialised label definitions from the bottom 2 label levels. The inclusion of more categories increased the number of target labels from 58 to 102. The *GMIL* category is ignored as it has only 5 documents. The data is summarised in Table 17 (the full Table can be found in the Appendix, Table 26) and the results are shown in Table 18.

As before, the LDA ensemble using mutual exclusion for the labels achieves the best 0/1-loss. However, as noted before 0/1-loss is very unforgiving, and does not necessarily reflect the requirements of the application scenario well. The per sample precision and recall metrics in Table 18 as well as the ranking loss give a more comprehensive picture of how the models compare with each other.

The two logistic regression ensembles and the LDA ensembles that do not take into account the label hierarchy are potentially simply making different precision/recall trade-offs (Figure 14). However, the multi-label per sample precision and recall in Table 18 are biased by large categories and do not account for differences in category importance or correlations between the categories. Since the categories lower down the hierarchy are

Model	0/1-loss	Precision	Recall	Ranking Loss
Decision Tree	0.940	0.404	0.325	0.572
Extra Trees	0.986	0.642	0.195	0.105
Random Forest	0.993	0.623	0.181	0.119
LR-OvA^{bow}	0.948	0.648	0.466	0.164
LR-OvAθ	0.952	0.773	0.346	0.087
LDA 100k	0.956	0.557	0.523	0.075
LDA H 100k	0.948	0.485	0.458	0.075
LDA H mutex 100k	0.851	0.638	0.443	0.075
LDA 800k	0.962	0.547	0.531	0.072
LDA H 800k	0.955	0.466	0.460	0.072
LDA H mutex 800k	0.851	0.650	0.447	0.072

Table 18: 0/1-loss, precision, recall and label ranking loss for all models on documents labelled with the 4 levels of the Reuters label hierarchy. For the 0/1-loss and label ranking loss lower is better. Precision and recall are measured as the average per document precision and recall. The comparison ensemble models have 200 estimators in them, and the topic based ensembles use a topic model with 200 topics.

more specialised than those at the top, the performance on the lower categories is critically important. Being able to separate articles about Politics from those about Sports is less useful[5] than being able to separate Football from Tennis, or UK Politics from German Politics.

Therefore, we also analysed how the models perform on each level of the label hierarchy, focussing on per category binary performance, averaged over all categories. For each category we measured the Precision, Recall and F1-score of all models, using anything not labelled as belonging to that category as the "other". The results are displayed in Tables 19 as averages over all label tiers and 20 for each label tier separately.

	Model	Precision	Recall	F1-score	Accuracy
All Labels					
	Logistic Regression OvA BoW	0.411	0.291	0.321	0.963
	Logistic Regression OvA θ	0.407	0.114	0.156	0.968
	LDA 100k	0.441	0.412	0.392	0.961
	LDA 800k	**0.452**	**0.427**	**0.403**	0.960

Table 19: Per category binary Precision, Recall, F1-score and Accuracy averaged across all label tiers (unweighted macro).

5 Relatively simple keyword matching should allow separating general high level categories from each other.

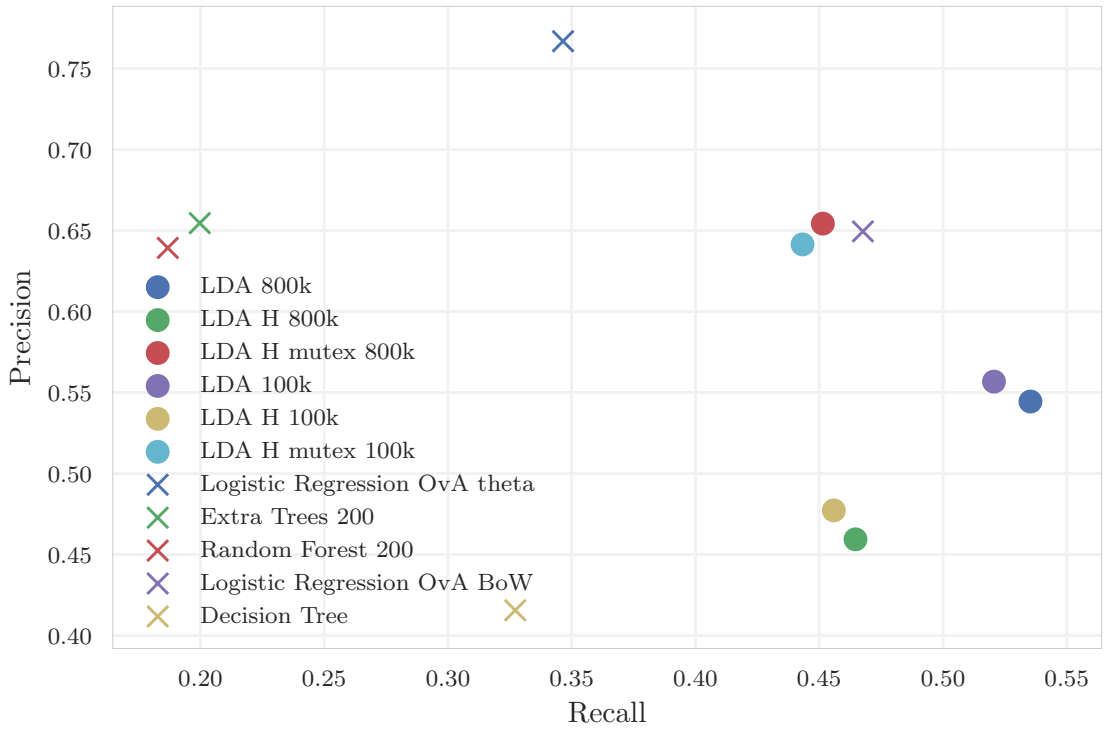


Figure 14: Precision-Recall trade-off for all models on the 4-tier label hierarchy.

Looking at the per category binary performance metrics over all tiers (Table 19) it is clear that both of the logistic regression ensembles suffer from the increased number of labels. Separating out each of the label tiers (Table 20) we see that the logistic regression ensembles perform better than the topical ensemble for the 4 1st-tier labels only. The topical ensembles achieve much higher precision and recall for labels lower down in the hierarchy (Table 20), Figure 15 shows the F1-score on each category against the size of the training set. The shaded areas show the quartiles of each score distribution. The logistic regression ensemble improves over the topical ensemble only for 3 of the largest categories. The topical ensemble is able to learn from much fewer training instances. This is noteworthy as the logistic regression ensemble that uses the document-topic vectors as the input signal does not match the performance of the topical ensemble of the BoW logistic regression ensemble. The improvement is not, therefore, simply a function of using a topical representation but a result of our ensemble model.

	Model	Precision	Recall	F1	Accuracy
Tier 1					
	Logistic Regression OvA BoW	0.748	0.702	0.722	0.827
	Logistic Regression OvA θ	0.847	0.668	0.725	0.846
	LDA 100k	0.716	0.674	0.689	0.798
	LDA 800k	0.709	0.661	0.679	0.791
Tier 2					
	Logistic Regression OvA BoW	0.406	0.268	0.304	0.962
	Logistic Regression OvA θ	0.431	0.098	0.144	0.967
	LDA 100k	0.452	0.412	0.401	0.961
	LDA 800k	0.449	0.431	0.410	0.960
Tier 3					
	Logistic Regression OvA BoW	0.387	0.280	0.303	0.977
	Logistic Regression OvA θ	0.331	0.078	0.115	0.981
	LDA 100k	0.399	0.385	0.352	0.975
	LDA 800k	0.431	0.399	0.367	0.974
Tier 4					
	Logistic Regression OvA BoW	0.377	0.398	0.368	0.980
	Logistic Regression OvA θ	0.610	0.234	0.311	0.986
	LDA 100k	0.487	0.505	0.473	0.984
	LDA 800k	0.503	0.503	0.486	0.985

Table 20: Per category binary Precision, Recall, F1-score and Accuracy for the different label tiers separated out. Note that the F1-score displayed is not the harmonic mean of the listed precision and recall values but the average of the individual F1-scores for each category.

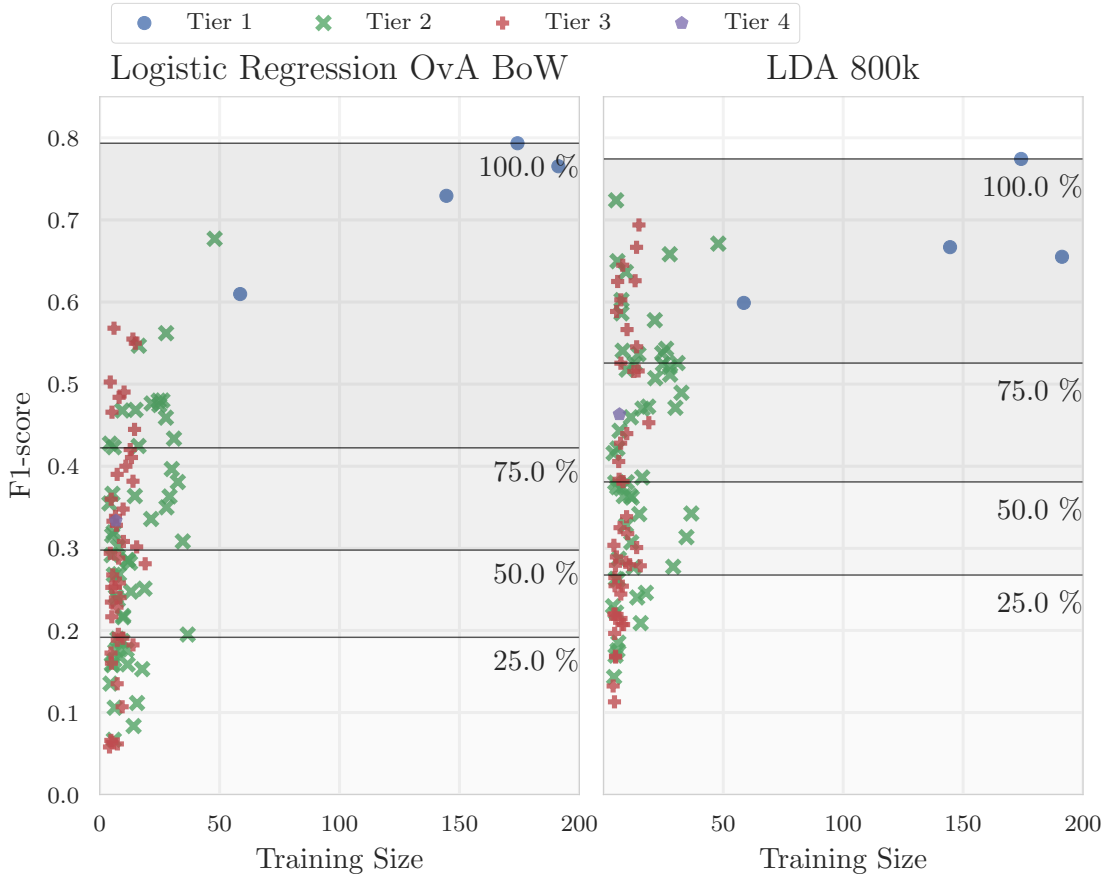


Figure 15: F1-score against training data size for the two best performing models. The shaded areas show the quartiles of the F1-score distribution for each model.

SECTION 4.5

Summary

We tested a novel topic based ensemble model against a number of multi-label classification algorithms on a complex classification task with 102 target labels organised in a hierarchy. The topical ensemble outperforms the comparison models on smaller categories where training data becomes scarce. Only when there are more than 100 training instances do the comparison models become competitive. Given the constraints of the application scenario, accumulating enough training data for the traditional models is problematic. Our model is able to handle a large number of target categories with a broad topical range. Our model also allows new labels to be added and old labels to be removed without impacting the performance of already learned categories.

5

CONCLUSIONS AND FUTURE WORK

In this final Chapter of the thesis we will first summarise our findings from Chapters 3 and then 4 and outline future research directions.

Many real world classification tasks deal with a corpus that has a broad topical range. These data sets can be difficult for standard classification algorithms as the class association of document features is subject to change based on topical context. In some cases a feature can reverse its class association between two topics. Our research focusses on the impact topical bias has on document classification and on finding ways to mitigate that. The central question we explore is: "*under what circumstances does resolving the topical context improve performance at a classification task?*".

Previous research[1] indicates that ensemble classifiers for sentiment analysis benefit from topical context. We selected two tasks that require a different understanding of the source text to be performed effectively: sentiment analysis and hierarchical multi-label document classification. The remaining two Sections summarise our empirical findings elaborate on the limitations of the research so far and outline future research.

SECTION 5.1

Topical Ensembles in Sentiment Classification

We showed that using an ensemble of classifiers for sentiment analysis does improve performance, especially when training data is scarce. However, we also showed that the topical bias of the ensemble model is unlikely to be the source of the improvement as a similar improvement is achieved by training an ensemble without topical bias. The individual SVM classifiers in the ensemble are not able to learn a better representation

1 Xiang and Zhou (2014) and Van Canneyt, Claeys, and Dhoedt (2015)

of the target classes using either a topically weighted training set or a data set that is sub-sampled based on topics. Contrary to previous research, our findings indicate that topical sub-sampling in fact leads to a worse model due to the sampling reducing training data further.

While the sub-sampling and instance weighting strategies are the immediately obvious methods for biasing the ensemble learning, there are alternatives that could be more effective. The document feature weights themselves could be weighted based on topic-term probabilities derived from the topic models or the features could be transformed based on discrete topic membership. Below we discuss both the limitations of our research thus far and outline future research directions.

DOCUMENT REPRESENTATION In all experiments we used a bag-of-words document representation and performed little feature engineering or selection. The bag-of-words document representation is limited in its ability to represent contextual information of words. This was part of the motivation for using a bag-of-words (BoW) model; to see if the contextual information inferred by a topic model could counterbalance the lack of context of the BoW model. Nevertheless, using a document representation that captures richer contextual information, for instance word vectors (Mikolov, Sutskever, et al., 2013), could improve model performance.

SAME DOCUMENT REPRESENTATION FOR TOPIC MODEL AND ENSEMBLE To ensure the alignment of the topical classifiers with the topic inferred from the topic model we used the same document representation across all models. There is good reason to believe that the topic model could benefit from a document representation that is different from that of the ensemble model and vice versa. For instance, including sentiment specific features for the ensemble model's document representation has been shown to be beneficial (Xiang and Zhou, 2014; Van Canneyt, Claeys, and Dhoedt, 2015). However, it is not clear how this would impact the correspondence between the document clusters from the topic model and the ensemble classifiers.

PAIRWISE CATEGORY COMPARISONS In order to keep the experimental methodology clear we considered pairs of categories in our experiments. This is a clear simplifi-

cation of the complexities that exist in real world corpora. Our empirical work did not demonstrate that the topical information specifically helps in sentiment classification task that involves data from two different topical areas. It is possible that this finding does not apply to cases where complex interactions between multiple different topical categories prevail.

NUMBER AND VARIABILITY OF DATASETS The empirical work focussed on user written product reviews in different topical categories. Sentiment analysis has been applied to many different kinds of data sources in addition to product reviews including social media posts, movie reviews and political opinion. The applicability of our results is limited to the product reviews domain. Previous research has shown that domain specific variations in how people express opinion do exist and that extra linguistic phenomena should in some cases be taken into account. Our results should therefore not be extrapolated to other domains without diligent comparisons between the regularities and irregularities of sentiment expression in those domains.

TOPIC-TERM WEIGHTING One possibility for creating a topically biased ensemble is to modify the feature values of each training instance for each local classifier based on the topic-term weights from the topic model. Each topic in the topic model is a probability distribution over the entire vocabulary. The probabilities could be used to weight the feature values in each training document and differentiate the training data between different topics that way. However, the per topic probability distributions assigned to the vocabulary tend to have a very long tail (Figure 16). Using the long tailed probability distribution as feature weights would essentially cause almost all of the document features to have a value very close to 0 as typically only the top 10 terms for any topic have weight above 0.05. At a minimum this approach would require very careful tuning.

DISCRETE TOPIC-TERM ASSIGNMENT Another alternative is to modify the feature space itself by assigning terms to topics based on some threshold. The document features would be transformed for instance with a topic ID suffix, replicating terms that belong to several topics. The threshold would need to be set high enough to create differentiation between the topic specific feature spaces, which would dramatically increase the sparsity

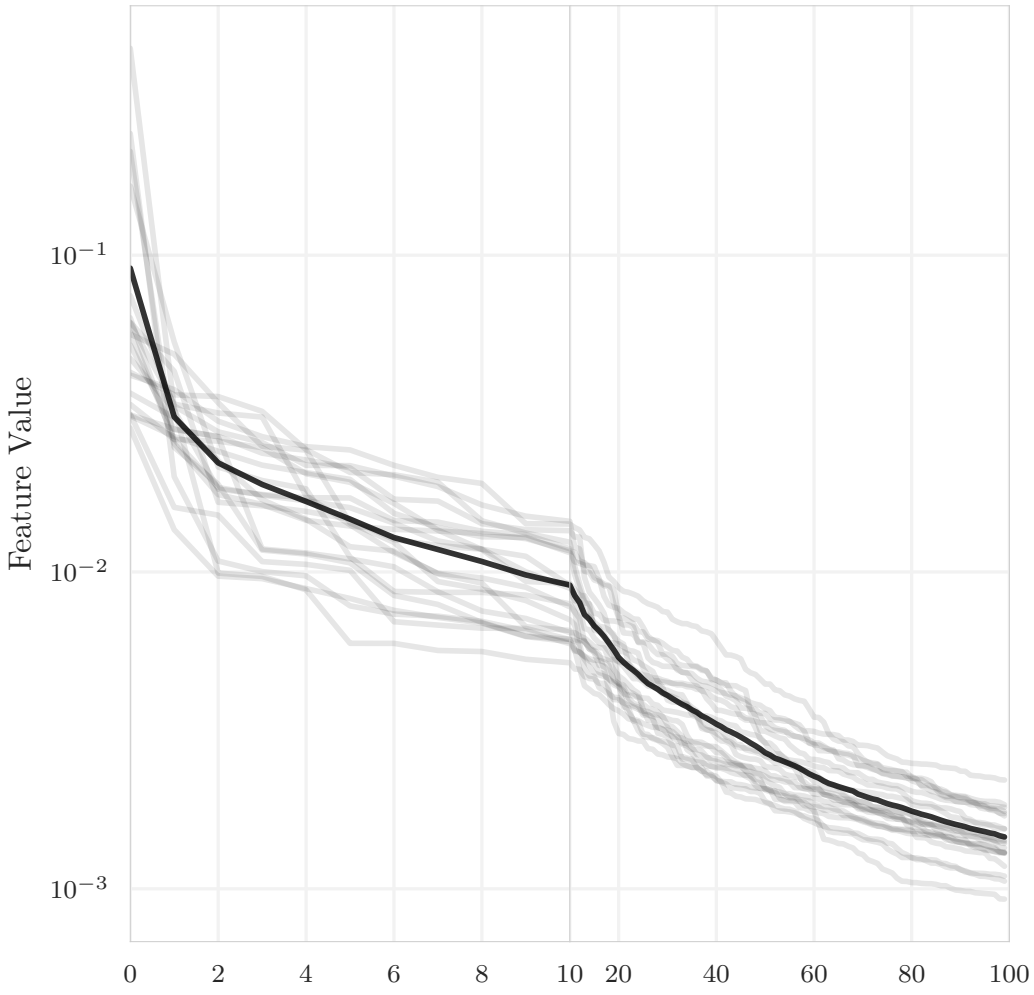


Figure 16: Top 100 topic term values for a 20 topic model. Thick black line is the mean. Each grey line is an individual topic.

of the document representation. Lowering the threshold would reduce the sparsity but also reduce the differentiation between the topic specific vocabularies.

THE CORRECT NUMBER OF TOPICS In all the experimental work we performed a parameter sweep for the best value of K, the number of topics. This is very expensive in practice. Having an automatic way of setting K would be preferable. There are a number of ways to evaluate the quality of a topic model for this purpose. The methods typically use information theoretic metrics measured on hold-out data. See Section 2.5.3.3 for an overview.

TOPIC SPECIFIC WORD EMBEDDINGS The bag-of-words document representation used in our experiments has known limitations. For instance, it only encodes the presence or absence of word identities and does not account for word semantics in a meaningful way. Neural Network models that use a continuous representation of word meaning have gained a lot of attention in recent years, including for document classification. The word embeddings are typically trained from a single source corpus and the embeddings do not discriminate between topics. A topic model could be used to train topic specific word embeddings that could then be fed into a neural network classifier.

FEATURE SELECTION In practical applications of sentiment classification, feature selection methods are often utilised to learn a better model. We did not include feature selection methods in our empirical work to maintain a strict relationship between the topic model and the local classifiers. However, as feature selection methods create radically different feature spaces for the classifier to operate on, exploring the implications of feature selection for topically rich corpora is an interesting area for future research.

SECTION 5.2

Topical Ensembles for Topical Content

We present a method for multi-label classification that is computationally efficient and outperforms strong baseline models. The benefit of our method is that it is able to utilise large amounts of unlabelled data to create a broad topical representation of a corpus. The topical representation is utilised to create an ensemble classifier. By taking into account the hierarchical nature of the label space our model improves over the state of the art.

A significant problem for developing better methods for multi-label classification is the difficulty in evaluating the models. No single metric completely captures the characteristics of a model and multiple different evaluations need to be performed to compare models with each other. Accounting for things such as label similarity and label prevalence in the evaluation itself is as of yet an unsolved issue. More work is needed to develop evaluation methodologies that account for these issues.

ALTERNATIVE TOPIC MODELS Our work demonstrates the value of distributed topical document representations for classification tasks. An open question is the extent to which the quality of the topic model impacts on the performance of the ensemble. Using a different topic model could yield significant further improvements in performance. Batmanghelich et al. (2016) for instance show how to produce more coherent topics than LDA. In the empirical work we trained a single topic model from a large unlabelled corpus. The label hierarchy could be utilised to train a hierarchy of topic models with increasing levels of granularity. How far this hierarchical approach could be pushed is an open question.

SEMI-SUPERVISED LEARNING One particular difficulty for deploying machine learning models in practice is the lack of high quality labelled data. The problem is exacerbated by an increase in the number of labels as each individual label requires a certain amount of training data. Our model could be used in a semi-supervised setting to label a large corpus which could then be used to improve the model.

NUMBER AND SPECIFICITY OF TARGET CATEGORIES We used the RCV1 corpus and its 103 topic categories organised in four hierarchical levels in our empirical work on multi-label classification. Some commonly used multi-label classification data sets, such as the Medical Subject Headings[2], contain many more target categories and many more levels in the hierarchy. The applicability of topic models is limited by their tendency to capture broad corpus wide patterns and overlook fine grained details. Although our method works better than the comparison models on the Reuters data it is important to better understand how these same patterns are reflected in other corpora that are labelled using different label hierarchies.

OTHER CLASSIFICATION SIGNALS Our model currently utilises a topic model and a set of weights to rank labels for new articles. The model however is very flexible and could integrate other signals in addition to the topic model. The topic model captures coarse grained structures in the data. The coarse grained structures could be amended

2 https://www.nlm.nih.gov/mesh/

by labelling named entities, which are likely to be very specific to certain labels. A way of combining these signals would then need to be developed.

WEIGHT SHARING New labels are continuously introduced in the application scenarios that motivated our model. All new labels will initially suffer from a cold start problem as very few documents will have been labelled for those categories. The distributed document representation derived using LDA allows similarity comparisons to be made between already existing categories and their relations with the topic model and the new documents. This offers the possibility to share weights between already learned categories and new, similar categories possibly reducing annotation costs and time to deployment for new labels.

Part III

APPENDIX

A

CHAPTER 3 - AMAZON DATA SETS

Table 21 lists the vocabulary agreement scores for the categories used in the experiments in Chapter 3.

Category Pair	Overlap	Agreement Score
Books - Cell Phones and Accessories	14268	0.7407
Movies and TV - Pet Supplies	15886	0.7783
CDs and Vinyl - Tools and Home Improvement	15779	0.8158
Baby - Digital Music	13829	0.8522
Baby - Home and Kitchen	13625	1.1029
Home and Kitchen - Office Products	14227	1.2060
Clothing Shoes and Jewelry - Sports and Outdoors	14672	1.2868
Digital Music - Musical Instruments	25174	1.3961

Table 21: Vocabulary agreement scores (see Section 3.2.1) and number of shared vocabulary items for the category pairs used in Chapter 3.

Category Pair	Overlap	Agreement Score
Books - Cell Phones and Accessories	14268	0.7407
Books - Pet Supplies	15825	0.7442
Cell Phones and Accessories - Movies and TV	14680	0.7485
Pet Supplies - Video Games	15490	0.7660
Books - Electronic	17085	0.7718
Movies and TV - Pet Supplies	15886	0.7783
Automotive - Books	14284	0.7826
Baby - Books	14274	0.7869
Cell Phones and Accessories - Video Games	15305	0.7873
Books - Tools and Home Improvement	15913	0.7880
Automotive - Movies and TV	14539	0.7888

Continued on next page

Category Pair	Overlap	Agreement Score
Books - Patio Lawn and Garden	16353	0.7940
Cell Phones and Accessories - Kindle Store	13964	0.7953
Amazon Instant Video - Cell Phones and Accessories	13160	0.7965
Electronic - Movies and TV	17651	0.8032
Baby - Video Games	14482	0.8036
Kindle Store - Pet Supplies	14888	0.8049
Baby - Movies and TV	14557	0.8064
Movies and TV - Patio Lawn and Garden	16415	0.8066
Automotive - Video Games	14706	0.8084
Amazon Instant Video - Pet Supplies	14033	0.8155
Movies and TV - Tools and Home Improvement	16194	0.8157
CDs and Vinyl - Tools and Home Improvement	15779	0.8158
Baby - Kindle Store	13556	0.8182
Amazon Instant Video - Automotive	12967	0.8195
Kindle Store - Tools and Home Improvement	15115	0.8204
Books - Health and Personal Ca	16015	0.8240
CDs and Vinyl - Cell Phones and Accessories	14415	0.8251
Health and Personal Ca - Movies and TV	16041	0.8255
Health and Personal Ca - Video Games	15583	0.8266
Digital Music - Pet Supplies	14840	0.8275
Patio Lawn and Garden - Video Games	16171	0.8280
Electronic - Kindle Store	16407	0.8285
Automotive - Kindle Store	13745	0.8295
Cell Phones and Accessories - Digital Music	14108	0.8307
CDs and Vinyl - Electronic	17445	0.8319
Books - Clothing Shoes and Jewelry	14262	0.8322
CDs and Vinyl - Pet Supplies	15354	0.8329
Movies and TV - Video Games	24806	0.8347
Kindle Store - Patio Lawn and Garden	15308	0.8360
Tools and Home Improvement - Video Games	16332	0.8369
Amazon Instant Video - Electronic	15476	0.8378
Health and Personal Ca - Kindle Store	15141	0.8401
Books - Home and Kitchen	15456	0.8404
Books - Video Games	23755	0.8429
Grocery and Gourmet Food - Video Games	15514	0.8465
Automotive - CDs and Vinyl	14240	0.8467
CDs and Vinyl - Patio Lawn and Garden	15924	0.8490
Baby - Digital Music	13829	0.8522
CDs and Vinyl - Health and Personal Ca	15644	0.8530
Amazon Instant Video - Patio Lawn and Garden	14530	0.8553
Books - Office Product	16331	0.8564
Amazon Instant Video - Tools and Home Improvement	14212	0.8566

Continued on next page

Category Pair	Overlap	Agreement Score
Automotive - Digital Music	13874	0.8571
Grocery and Gourmet Food - Movies and TV	16646	0.8602
Electronic - Video Games	18295	0.8627
Books - Grocery and Gourmet Food	16866	0.8643
Amazon Instant Video - Health and Personal Ca	14271	0.8661
Kindle Store - Video Games	21620	0.8684
Amazon Instant Video - Video Games	20773	0.8695
CDs and Vinyl - Video Games	23274	0.8698
Digital Music - Patio Lawn and Garden	15525	0.8730
Clothing Shoes and Jewelry - Movies and TV	14553	0.8747
Amazon Instant Video - Baby	12895	0.8763
Cell Phones and Accessories - Pet Supplies	12765	0.8775
Home and Kitchen - Movies and TV	15437	0.8779
Digital Music - Electronic	16877	0.8787
Baby - CDs and Vinyl	14329	0.8800
Clothing Shoes and Jewelry - Video Games	14195	0.8826
Beauty - Books	14742	0.8850
Books - Sports and Outdoo	17198	0.8863
Grocery and Gourmet Food - Kindle Store	15666	0.8870
Digital Music - Tools and Home Improvement	15382	0.8879
Cell Phones and Accessories - Health and Personal Ca	13225	0.8957
Home and Kitchen - Kindle Store	14477	0.8967
Amazon Instant Video - Office Product	14648	0.8968
Digital Music - Health and Personal Ca	15266	0.8971
Home and Kitchen - Video Games	15079	0.8981
Clothing Shoes and Jewelry - Kindle Store	13837	0.8994
Amazon Instant Video - Home and Kitchen	13710	0.8997
Cell Phones and Accessories - Grocery and Gourmet Food	12162	0.9000
Electronic - Pet Supplies	14257	0.9003
Beauty - Movies and TV	14944	0.9072
Movies and TV - Office Product	16421	0.9072
Beauty - Video Games	14209	0.9123
Beauty - CDs and Vinyl	14859	0.9123
Amazon Instant Video - Clothing Shoes and Jewelry	13097	0.9132
Apps for Android - Books	12912	0.9165
CDs and Vinyl - Grocery and Gourmet Food	16289	0.9177
Baby - Cell Phones and Accessoriesries	12685	0.9196
Amazon Instant Video - Grocery and Gourmet Food	14840	0.9222
Amazon Instant Video - Beauty	13409	0.9227
Kindle Store - Movies and TV	26092	0.9228
Digital Music - Home and Kitchen	14667	0.9231
Cell Phones and Accessories - Patio Lawn and Garden	13419	0.9248

Continued on next page

Category Pair	Overlap	Agreement Score
Movies and TV - Sports and Outdoors	17591	0.9262
CDs and Vinyl - Clothing Shoes and Jewelry	14347	0.9276
CDs and Vinyl - Home and Kitchen	14993	0.9282
Amazon Instant Video - Sports and Outdoors	15457	0.9285
Pet Supplies - Tools and Home Improvement	14443	0.9304
Clothing Shoes and Jewelry - Digital Music	13902	0.9313
Books - Movies and TV	30004	0.9318
Cell Phones and Accessories - Tools and Home Improvement	14065	0.9330
Office Product - Video Games	16751	0.9330
Apps for Android - Movies and TV	13057	0.9349
Kindle Store - Office Products	15703	0.9357
Book - CDs and Vinyl	26908	0.9364
Beauty - Kindle Store	14094	0.9370
Health and Personal Care - Pet Supplies	15046	0.9428
Sports and Outdoors - Video Games	17401	0.9437
Electronics - Grocery and Gourmet Food	13672	0.9444
Electronics - Health and Personal Care	14764	0.9523
Automotive - Pet Supplies	13708	0.9540
CDs and Vinyl - Kindle Store	24118	0.9545
CDs and Vinyl - Sports and Outdoors	17185	0.9552
Baby - Electronics	13849	0.9558
Automotive - Cell Phones and Accessories	13253	0.9559
CDs and Vinyl - Office Products	16140	0.9570
Baby - Grocery and Gourmet Food	12740	0.9586
Apps for Android - Video Games	13431	0.9605
Health and Personal Care - Tools and Home Improvement	14614	0.9612
Kindle Store - Sports and Outdoors	16314	0.9615
Grocery and Gourmet Food - Pet Supplies	14259	0.9638
Digital Music - Video Games	22715	0.9646
Digital Music - Grocery and Gourmet Food	15858	0.9650
Amazon Instant Video - Book	23992	0.9650
Amazon Instant Video - CDs and Vinyl	23064	0.9672
Patio Lawn and Garden - Pet Supplies	15287	0.9676
Amazon Instant Video - Movies and TV	25068	0.9684
Digital Music - Office Product	15566	0.9708
Book - Kindle Store	26952	0.9741
Cell Phones and Accessories - Electronic	16034	0.9744
Amazon Instant Video - Kindle Store	22153	0.9745
Grocery and Gourmet Food - Tools and Home Improvement	13390	0.9760
Baby - Health and Personal Care	13828	0.9774
CDs and Vinyl - Movies and TV	28850	0.9796
Baby - Pet Supplies	13967	0.9801

Continued on next page

Category Pair	Overlap	Agreement Score
Automotive - Electronics	14673	0.9813
Electronic - Patio Lawn and Garden	15183	0.9817
Automotive - Baby	13080	0.9833
Automotive - Health and Personal Care	13813	0.9843
Baby - Patio Lawn and Garden	13911	0.9866
Automotive - Tools and Home Improvement	15207	0.9881
Beauty - Cell Phones and Accessories	12059	0.9884
Health and Personal Care - Patio Lawn and Garden	14808	0.9891
Electronics - Tools and Home Improvement	16071	0.9926
Office Product - Pet Supplies	13803	0.9940
Automotive - Grocery and Gourmet Food	12674	0.9945
Apps for Android - Pet Supplies	10659	0.9948
Apps for Android - Cell Phones and Accessories	11230	0.9971
Baby - Tools and Home Improvement	13881	0.9973
Grocery and Gourmet Food - Patio Lawn and Garden	14275	0.9978
Automotive - Patio Lawn and Garden	14722	1.0049
Amazon Instant Video - Apps for Android	12279	1.0050
Apps for Android - CDs and Vinyl	12795	1.0067
Clothing Shoes and Jewelry - Pet Supplies	12717	1.0093
Apps for Android - Kindle Store	12829	1.0107
Beauty - Digital Music	14496	1.0110
Cell Phones and Accessories - Sports and Outdoors	14108	1.0112
Apps for Android - Health and Personal Care	10923	1.0128
Apps for Android - Tools and Home Improvement	11066	1.0151
Cell Phones and Accessories - Home and Kitchen	13051	1.0160
Digital Music - Sports and Outdoors	16682	1.0169
Cell Phones and Accessories - Clothing Shoes and Jewelry	12516	1.0174
Amazon Instant Video - Digital Music	21845	1.0204
Grocery and Gourmet Food - Health and Personal Care	14920	1.0207
Apps for Android - Digital Music	12493	1.0242
Digital Music - Movies and TV	26804	1.0279
Book - Digital Music	25446	1.0282
Apps for Android - Baby	10219	1.0311
Apps for Android - Patio Lawn and Garden	10974	1.0324
Grocery and Gourmet Food - Office Products	13364	1.0353
Home and Kitchen - Pet Supplies	14320	1.0394
Automotive - Office Products	13967	1.0407
Clothing Shoes and Jewelry - Patio Lawn and Garden	13109	1.0415
Beauty - Pet Supplies	13490	1.0416
Digital Music - Kindle Store	22778	1.0419
Patio Lawn and Garden - Tools and Home Improvement	16011	1.0428
Beauty - Electronics	13268	1.0431

Continued on next page

Category Pair	Overlap	Agreement Score
Clothing Shoes and Jewelry - Tools and Home Improvement	13282	1.0433
Apps for Android - Automotive	10430	1.0465
Cell Phones and Accessories - Office Products	14704	1.0492
Pet Supplies - Sports and Outdoors	15086	1.0527
Clothing Shoes and Jewelry - Electronics	13704	1.0528
Apps for Android - Grocery and Gourmet Food	10694	1.0568
Baby - Beauty	12616	1.0596
Beauty - Patio Lawn and Garden	13419	1.0604
Health and Personal Care - Home and Kitchen	14452	1.0607
Clothing Shoes and Jewelry - Grocery and Gourmet Food	12417	1.0646
Office Products - Patio Lawn and Garden	14585	1.0657
Cell Phones and Accessories - Musical Instruments	15603	1.0680
Clothing Shoes and Jewelry - Health and Personal Care	13260	1.0687
Beauty - Tools and Home Improvement	13129	1.0689
Health and Personal Care - Office Products	14293	1.0714
Baby - Office Products	13406	1.0715
Apps for Android - Electronics	12385	1.0722
Electronics - Sports and Outdoors	16177	1.0728
Automotive - Beauty	12632	1.0729
Baby - Clothing Shoes and Jewelry	12691	1.0735
CDs and Vinyls - Musical Instruments	27285	1.0779
Amazon Instant Video - Musical Instruments	20835	1.0806
Electronics - Home and Kitchen	14470	1.0830
Books - Toys and Games	19406	1.0830
Office Products - Tools and Home Improvement	15341	1.0834
Automotive - Clothing Shoes and Jewelry	12700	1.0856
Automotive - Home and Kitchen	13831	1.0901
Sports and Outdoors - Tools and Home Improvement	16170	1.0943
CDs and Vinyls - Digital Music	29016	1.0971
Grocery and Gourmet Food - Home and Kitchen	14526	1.1021
Automotive - Musical Instruments	15455	1.1025
Baby - Home and Kitchen	13625	1.1029
Automotive - Sports and Outdoors	15143	1.1032
Baby - Sports and Outdoors	14525	1.1033
Musical Instruments - Pet Suppl	15692	1.1037
Cell Phones and Accessories - Toys and Games	14031	1.1073
Apps for Android - Clothing Shoes and Jewelry	10218	1.1094
Apps for Android - Beauty	10132	1.1109
Beauty - Grocery and Gourmet Food	13757	1.1149
Movies and TV - Toys and Games	19599	1.1185
Grocery and Gourmet Food - Sports and Outdoors	14493	1.1187
Electronics - Office Products	16831	1.1206

Continued on next page

Category Pair	Overlap	Agreement Score
Automotive - Toys and Games	14086	1.1212
Home and Kitchen - Tools and Home Improvement	14988	1.1275
Health and Personal Care - Sports and Outdoors	15641	1.1276
Home and Kitchen - Patio Lawn and Garden	15268	1.1281
Baby - Musical Instruments	14742	1.1310
Movies and TV - Musical Instruments	25424	1.1316
CDs and Vinyls - Toys and Games	18927	1.1321
Books - Musical Instruments	24561	1.1342
Apps for Android - Home and Kitchen	10669	1.1353
Patio Lawn and Garden - Sports and Outdoors	16103	1.1376
Beauty - Home and Kitchen	13238	1.1384
Amazon Instant Video - Toys and Games	17361	1.1396
Apps for Android - Sports and Outdoors	11604	1.1434
Apps for Android - Office Product	11868	1.1447
Musical Instruments - Video Games	22115	1.1453
Electronic - Toys and Games	16023	1.1480
Grocery and Gourmet Food - Musical Instruments	15995	1.1494
Musical Instruments - Tools and Home Improvement	17070	1.1540
Toys and Games - Video Games	19608	1.1559
Kindle Store - Toys and Games	18251	1.1571
Clothing Shoes and Jewelry - Home and Kitchen	12780	1.1598
Beauty - Office Products	12978	1.1612
Health and Personal Care - Musical Instruments	16102	1.1631
Grocery and Gourmet Food - Toys and Games	14530	1.1635
Electronics - Musical Instruments	18980	1.1666
Clothing Shoes and Jewelry - Office Products	13216	1.1690
Beauty - Clothing Shoes and Jewelry	12519	1.1700
Beauty - Health and Personal Care	14699	1.1765
Health and Personal Care - Toys and Games	14847	1.1792
Digital Music - Toys and Games	18106	1.1847
Pet Supplies - Toys and Games	14889	1.1858
Musical Instruments - Patio Lawn and Garden	16672	1.1991
Home and Kitchen - Office Products	14227	1.2060
Tools and Home Improvement - Toys and Games	15407	1.2090
Office Product - Sports and Outdoors	15217	1.2102
Home and Kitchen - Sports and Outdoors	15155	1.2119
Apps for Android - Musical Instruments	12885	1.2139
Patio Lawn and Garden - Toys and Games	15459	1.2193
Home and Kitchen - Musical Instruments	15758	1.2301
Apps for Android - Toys and Games	12269	1.2393
Baby - Toys and Games	14582	1.2396
Beauty - Sports and Outdoors	14149	1.2462

Continued on next page

Category Pair	Overlap	Agreement Score
Clothing Shoes and Jewelry - Musical Instruments	14879	1.2524
Clothing Shoes and Jewelry - Toys and Games	13876	1.2650
Home and Kitchen - Toys and Games	14764	1.2688
Kindle Store - Musical Instruments	22520	1.2801
Beauty - Toys and Games	13646	1.2852
Clothing Shoes and Jewelry - Sports and Outdoors	14672	1.2868
Musical Instrument - Office Products	17218	1.3265
Office Product - Toys and Games	15576	1.3575
Sports and Outdoo - Toys and Games	16478	1.3686
Beauty - Musical Instruments	15109	1.3895
Digital Music - Musical Instruments	25174	1.3961
Musical Instruments - Sports and Outdoors	17901	1.4398
Musical Instruments - Toys and Games	18752	1.6403

Table 22: Vocabulary agreement scores (see Section 3.2.1) and number of shared vocabulary items for the category pairs used in Chapter 3.

B

CHAPTER 3 - BALANCED DATA, NO WEIGHT SCALING

Table 23 lists the full results for the topical ensemble for sentiment classification on a balanced dataset with weight scaling and no tie braking.

		SVM	LDA+SVM		
Vote Aggregation			0/1	θ	†
Topics	Weight Scaling				
1	-	0.675			
	0/1		0.675	0.675	
	0/2		0.674	0.674	
	1/2		0.674	0.674	
	1/4		0.673	0.673	
	2/3		0.674	0.674	
	3/4		0.674	0.674	
	1/10		0.671	0.671	
2	0/1		0.655	0.675	0.624
	0/2		0.654	0.674	0.622
	1/2		0.675	0.679	0.674
	1/4		0.671	0.678	0.666
	2/3		0.675	0.676	0.675

Continued on next page

		SVM	LDA+SVM		
Vote Aggregation			0/1	θ	†
Topics	Weight Scaling				
	3/4		0.675	0.676	0.675
	1/10		0.663	0.674	0.650
3	0/1		0.662	0.670	
	0/2		0.661	0.669	
	1/2		0.677	0.678	
	1/4		0.675	0.678	
	2/3		0.676	0.677	
	3/4		0.677	0.676	
	1/10		0.670	0.672	
4	0/1		0.658	0.666	
	0/2		0.657	0.665	
	1/2		0.678	0.678	
	1/4		0.676	0.677	
	2/3		0.677	0.676	
	3/4		0.678	0.675	
	1/10		0.672	0.672	
5	0/1		0.665	0.666	
	0/2		0.664	0.664	
	1/2		0.678	0.677	
	1/4		0.676	0.677	
	2/3		0.678	0.676	
	3/4		0.677	0.675	
	1/10		0.674	0.672	

Continued on next page

		SVM	LDA+SVM	
Vote Aggregation			0/1	θ	†
Topics	Weight Scaling			
6	0/1		0.664	0.662	
	0/2		0.664	0.663	
	1/2		0.678	0.678	
	1/4		0.677	0.678	
	2/3		0.678	0.677	
	3/4		0.679	0.676	
	1/10		0.674	0.672	
8	0/1		0.667	0.662	
	0/2		0.666	0.661	
	1/2		0.679	0.678	
	1/4		0.678	0.677	
	2/3		0.679	0.676	
	3/4		0.679	0.676	
	1/10		0.676	0.673	
10	0/1		0.667	0.659	
	0/2		0.666	0.660	
	1/2		0.679	0.678	
	1/4		0.679	0.678	
	2/3		0.678	0.677	
	3/4		0.678	0.677	
	1/10		0.677	0.674	
20	0/1		0.667	0.657	
	0/2		0.666	0.656	

Continued on next page

		SVM	LDA+SVM		
Vote Aggregation			0/1	θ	†
Topics	Weight Scaling				
	1/2		0.679	0.679	
	1/4		0.679	0.680	
	2/3		0.678	0.677	
	3/4		0.678	0.676	
	1/10		0.678	0.677	
30	0/1		0.665	0.652	
	0/2		0.665	0.651	
	1/2		0.680	0.679	
	1/4		0.679	0.679	
	2/3		0.679	0.677	
	3/4		0.679	0.677	
	1/10		0.679	0.677	
40	0/1		0.664	0.652	
	0/2		0.664	0.651	
	1/2		0.679	0.678	
	1/4		0.679	0.679	
	2/3		0.679	0.677	
	3/4		0.679	0.676	
	1/10		0.679	0.678	

Table 23: Mathews Correlation Coefficient for balanced data (dataset (a), see Section 3.2 Table 3.

C

CHAPTER 3 - SUB-SAMPLE RESULTS

Table 24 lists the results for sub-sampling the training data for an ensemble classifier, instead of using training data weights.

	SVM	LDA+SVM	
Vote Aggregation		0/1	θ
Topics			
(a) 2500 2500 2500 2500			
2	0.674	0.674	0.673
3	0.674	0.675	0.673
4	0.674	0.674	0.671
5	0.674	0.671	0.669
6	0.674	0.669	0.667
8	0.674	0.663	0.662
10	0.674	0.658	0.654
20	0.674	0.645	0.644
30	0.674	0.640	0.636
40	0.674	0.637	0.633
(b) 500 500 4500 4500			
2	0.682	0.681	0.681

Continued on next page

	SVM	LDA+SVM	
Vote Aggregation		0/1	θ
Topics			
3	0.682	0.682	0.681
4	0.682	0.680	0.678
5	0.682	0.679	0.674
6	0.682	0.678	0.673
8	0.682	0.672	0.669
10	0.682	0.668	0.664
20	0.682	0.657	0.652
30	0.682	0.652	0.649
40	0.682	0.648	0.644
(c) 500 4500 4500 500			
2	0.812	0.810	0.811
3	0.812	0.813	0.810
4	0.812	0.812	0.810
5	0.812	0.811	0.806
6	0.812	0.810	0.806
8	0.812	0.807	0.803
10	0.812	0.805	0.801
20	0.812	0.798	0.794
30	0.812	0.795	0.791
40	0.812	0.794	0.790
(d) 4500 500 4500 500			
2	0.484	0.472	0.478
3	0.484	0.487	0.474

Continued on next page

	SVM	LDA+SVM	
Vote Aggregation		0/1	θ
Topics			
4	0.484	0.480	0.476
5	0.484	0.486	0.473
6	0.484	0.475	0.468
8	0.484	0.463	0.459
10	0.484	0.449	0.450
20	0.484	0.429	0.428
30	0.484	0.412	0.423
40	0.484	0.406	0.410
(e) 500 1000 8000 500			
2	0.670	0.664	0.664
3	0.670	0.673	0.663
4	0.670	0.671	0.666
5	0.670	0.673	0.661
6	0.670	0.669	0.659
8	0.670	0.666	0.656
10	0.670	0.658	0.653
20	0.670	0.647	0.639
30	0.670	0.640	0.637
40	0.670	0.636	0.632
(f) 8000 1000 250 750			
2	0.628	0.623	0.626
3	0.628	0.632	0.625
4	0.628	0.624	0.625

Continued on next page

	SVM	LDA+SVM	
Vote Aggregation		0/1	θ
Topics			
5	0.628	0.631	0.623
6	0.628	0.622	0.621
8	0.628	0.616	0.613
10	0.628	0.607	0.610
20	0.628	0.595	0.597
30	0.628	0.588	0.596
40	0.628	0.583	0.589

Table 24: Mathews Correlation Coefficient for sub-sampled ensemble training.

D

CHAPTER 4 - CATEGORY COUNTS

Topic Code	Size	Explanation	Depth	Parent
CCAT	374316	Corporate/Industrial	1	
C11	24325	Strategy/Plans	2	CCAT
C12	11944	Legal/Judicial	2	CCAT
C13	37410	Regulation/Policy	2	CCAT
C14	7410	Share Listings	2	CCAT
C15	150164	Performance	2	CCAT
C151	81875	Accounts/Earnings	3	C15
C1511	23212	Annual Results	4	C151
C152	73092	Comment/Forecasts	3	C15
C16	1920	Insolvency/Liquidity	3	CCAT
C17	41829	Funding/Capital	2	CCAT
C171	18313	Share Capital	3	C17
C172	11487	Bonds/Debt Issues	3	C17
C173	2636	Loans/Credits	3	C17
C174	5871	Credit Ratings	3	C17
C18	51480	Ownership Changes	2	CCAT
C181	43374	Mergers/Acquisitions	3	C18
C182	4671	Asset Transfers	3	C18
C183	7406	Privatisations	3	C18
C21	25403	Production/Services	2	CCAT
C22	6119	New products/Services	2	CCAT
C23	2625	Research/Development	2	CCAT
C24	32153	Capacity/Facilities	2	CCAT
C31	40506	Markets/Marketing	2	CCAT
C311	4299	Domestic Markets	3	C31
C312	6648	External Markets	3	C31
C313	1115	Market Share	3	C31
C32	2084	Advertising/Promotion	2	CCAT
C33	15331	Contracts/Orders	2	CCAT

Continued on next page

Topic Code	Size	Explanation	Depth	Parent
C331	1210	Defence Contracts	3	C33
C34	4835	Monopolies/Competition	2	CCAT
C41	11354	Management	2	CCAT
C411	10272	Management Moves	3	C41
C42	11878	Labour	2	CCAT
ECAT	117539	Economics	1	
E11	8568	Economic Performance	2	ECAT
E12	27078	Monetary/Economic	2	ECAT
E121	2182	Money Supply	3	E12
E13	6345	Inflation/Prices	2	ECAT
E131	5659	Consumer Prices	3	E13
E132	939	Wholesale Prices	3	E13
E14	2086	Consumer Finance	2	ECAT
E141	376	Personal Income	3	E14
E142	200	Consumer Credit	3	E14
E143	1206	Retail Sales	3	E14
E21	43128	Government Finance	2	ECAT
E211	15768	Expenditure/Revenue	3	E21
E212	27405	Government Borrowing	3	E21
E31	2342	Output/Capacity	2	ECAT
E311	1701	Industrial Production	3	E31
E312	52	Capacity Utilization	2	E31
E313	111	Inventories	3	E31
E41	16900	Employment/Labour	2	ECAT
E411	2136	Unemployment	3	E41
E51	20722	Trade/Reserves	2	ECAT
E511	2933	Balance of Payments	3	E51
E512	12634	Merchandise Trade	3	E51
E513	2290	Reserves	3	E51
E61	391	Housing Starts	2	ECAT
E71	5270	Leading Indicators	2	ECAT
GCAT	234873	Government/Social	1	
G15	19152	European Community	2	GCAT
G151	3307	EC Internal Market	3	G15
G152	2107	EC Corporate Policy	3	G15
G153	2360	EC Agriculture Policy	3	G15
G154	8404	EC Monetary/Economic	3	G15
G155	2124	EC Institutions	3	G15
G156	260	EC Environment Issues	3	G15
G157	2036	EC Competition/Subsidy	3	G15
G158	4300	EC External Relations	3	G15
G159	40	EC General	3	G15
GCRIM	32219	Crime, Law Enforcement	2	GCAT

Continued on next page

Topic Code	Size	Explanation	Depth	Parent
GDEF	8842	Defence	2	GCAT
GDIP	37739	International Relations	2	GCAT
GDIS	8657	Disasters and Accidents	2	GCAT
GENT	3801	Arts, Culture, Entertainment	2	GCAT
GENV	6261	Environment and Natural World	2	GCAT
GFAS	313	Fashion	2	GCAT
GHEA	6030	Health	2	GCAT
GJOB	17241	Labour Issues	2	GCAT
GMIL	5	Millennium Issues	2	GCAT
GOBIT	844	Obituaries	2	GCAT
GODD	2802	Human Interest	2	GCAT
GPOL	56878	Domestic Politics	2	GCAT
GPRO	5498	Biographies, Personalities, People	2	GCAT
GREL	2849	Religion	2	GCAT
GSCI	2410	Science and Technology	2	GCAT
GSPO	35317	Sports	2	GCAT
GTOUR	680	Travel and Tourism	2	GCAT
GVIO	32615	War, Civil War	2	GCAT
GVOTE	11532	Elections	2	GCAT
GWEA	3878	Weather	2	GCAT
GWELF	1869	Welfare, Social Services	2	GCAT
MCAT	200190	Markets	1	
M11	48700	Equity Markets	2	MCAT
M12	26036	Bond Markets	2	MCAT
M13	52972	Money Markets	2	MCAT
M131	28185	Interbank Markets	3	M13
M132	26752	Forex Markets	3	M13
M14	85100	Commodity Markets	2	MCAT
M141	47708	Soft Commodities	3	M14
M142	12136	Metals Trading	3	M14
M143	21957	Energy Markets	3	M14

Table 25: Topic codes, category sizes and the topic code explanation for every topic code in RCV1.

Table 26 lists the average training and test size counts for all categories used in Chapter 4.

Category	Training Size	Test Size	Total	Depth
CCAT	189.88	759.12	374316	1

Continued on next page ...

Category	Training Size	Test Size	Total	Depth
C11	13.20	53.68	24325	2
C12	10.12	43.12	11944	2
C13	36.80	142.64	37410	2
C14	6.16	24.24	7410	2
C15	25.56	105.92	150164	2
C151	13.32	54.24	81875	3
C1511	6.60	25.36	23212	4
C152	13.20	55.56	73092	3
C16	5.12	19.56	1920	2
C17	30.04	125.60	41829	2
C171	10.40	42.64	18313	3
C172	5.72	24.24	11487	3
C173	4.68	19.32	2636	3
C174	5.56	24.04	5871	3
C18	27.60	111.56	51480	2
C181	19.04	78.40	43374	3
C182	5.64	20.24	4671	3
C183	6.28	26.76	7406	3
C21	15.80	59.12	25403	2
C22	6.60	24.20	6119	2
C23	5.44	23.60	2625	2
C24	19.36	69.52	32153	2
C31	35.96	141.52	40506	2
C311	6.80	32.16	4299	3
C312	10.20	36.80	6648	3
C313	4.76	17.84	1115	3
C32	5.32	21.88	2084	2
C33	12.72	48.52	15331	2
C331	4.88	19.32	1210	3
C34	11.52	44.16	4835	2
C41	9.88	42.20	11354	2
C411	9.20	37.32	10272	3
C42	14.12	57.88	11878	2
ECAT	143.08	572.68	117539	1
E11	12.12	47.40	8568	2
E12	30.04	120.68	27078	2
E121	5.72	21.84	2182	3
E13	14.08	57.88	6345	2
E131	10.16	42.32	5659	3
E132	5.08	19.60	939	3
E14	15.68	67.44	2086	2
E141	4.96	20.48	376	3

Continued on next page . . .

Category	Training Size	Test Size	Total	Depth
E142	4.52	17.80	200	3
E143	7.00	29.52	1206	3
E21	27.44	109.16	43128	2
E211	14.64	57.68	15768	3
E212	12.76	51.20	27405	3
E31	17.76	63.24	2342	2
E311	10.96	38.12	1701	3
E312	4.68	15.96	52	3
E313	4.44	16.92	111	3
E41	24.28	101.96	16900	2
E411	6.12	24.68	2136	3
E51	28.04	114.64	20722	2
E511	7.24	31.08	2933	3
E512	14.28	59.12	12634	3
E513	5.32	21.24	2290	3
E61	4.36	16.44	391	2
E71	4.44	18.20	5270	2
GCAT	170.92	680.60	234873	1
G15	46.84	188.68	19152	2
G151	7.20	33.16	3307	3
G152	9.24	36.20	2107	3
G153	7.08	28.48	2360	3
G154	14.64	59.16	8404	3
G155	7.24	29.28	2124	3
G156	4.68	18.20	260	3
G157	8.04	32.16	2036	3
G158	9.88	39.24	4300	3
G159	4.12	16.04	40	3
GCRIM	20.68	87.24	32219	2
GDEF	9.64	35.92	8842	2
GDIP	20.56	80.24	37739	2
GDIS	8.60	33.96	8657	2
GENT	7.64	30.00	3801	2
GENV	12.68	46.00	6261	2
GFAS	4.80	16.72	313	2
GHEA	9.32	40.44	6030	2
GJOB	24.44	101.48	17241	2
GOBIT	5.56	21.16	844	2
GODD	5.88	21.76	2802	2
GPOL	33.60	125.36	56878	2
GPRO	11.80	45.04	5498	2
GREL	5.84	20.28	2849	2

Continued on next page ...

Category	Training Size	Test Size	Total	Depth
GSCI	5.20	21.80	2410	2
GSPO	5.32	20.00	35317	2
GTOUR	4.96	18.56	680	2
GVIO	12.24	48.84	32615	2
GVOTE	7.84	28.20	11532	2
GWEA	4.52	21.96	3878	2
GWELF	5.80	21.08	1869	2
MCAT	58.84	241.76	200190	1
M11	8.20	34.92	48700	2
M12	7.68	31.32	26036	2
M13	19.48	84.40	52972	2
M131	7.88	37.76	28185	3
M132	12.80	51.04	26752	3
M14	29.00	114.80	85100	2
M141	15.80	61.12	47708	3
M142	6.32	26.76	12136	3
M143	7.84	29.04	21957	3

Table 26: Average category counts for the 4-level hierarchy over 25 random samples.

E

CHAPTER 4 - PER CATEGORY PERFORMANCE

Table 27 lists the per category performance metrics (Precision, Recall, F1-score and Accuracy) for each label tier.

Model	Precision	Recall	F1	Accuracy
All Labels				
Decision Tree	0.255	0.180	0.193	0.953
Extra Trees	0.282	0.033	0.049	0.964
Random Forest 200	0.192	0.024	0.035	0.964
LR-OvA^{bow}	0.411	0.291	0.321	0.963
LR-OvA$^{\theta}$	0.407	0.114	0.156	0.968
LDA 100k	0.441	0.412	0.392	0.961
LDA H 100k	0.418	0.315	0.319	0.961
LDA H mutex 100k	0.461	0.277	0.304	0.965
LDA 800k	0.452	0.427	0.403	0.960
LDA H 800k	0.424	0.323	0.324	0.960
LDA H mutex 800k	0.469	0.277	0.311	0.966
Tier 1				
Decision Tree	0.642	0.528	0.573	0.753
Extra Trees	0.886	0.418	0.502	0.797
Random Forest	0.814	0.396	0.477	0.788
LR-OvA^{bow}	0.748	0.702	0.722	0.827
LR-OvAθ	0.847	0.668	0.725	0.846

Continued on next page ...

Model	Precision	Recall	F1	Accuracy
LDA 100k	0.716	0.674	0.689	0.798
LDA H 100k	0.715	0.675	0.689	0.798
LDA H mutex 100k	0.864	0.622	0.719	0.838
LDA 800k	0.709	0.661	0.679	0.791
LDA H 800k	0.705	0.663	0.678	0.790
LDA H mutex 800k	0.864	0.627	0.723	0.840
Tier 2				
Decision Tree	0.228	0.155	0.169	0.952
Extra Trees	0.235	0.011	0.020	0.964
Random Forest	0.159	0.006	0.011	0.964
LR-OvA^{bow}	0.406	0.268	0.304	0.962
LR-OvA$^{\theta}$	0.431	0.098	0.144	0.967
LDA 100k	0.452	0.412	0.401	0.961
LDA H 100k	0.432	0.343	0.348	0.960
LDA H mutex 100k	0.502	0.261	0.304	0.965
LDA 800k	0.449	0.431	0.410	0.960
LDA H 800k	0.427	0.349	0.351	0.959
LDA H mutex 800k	0.504	0.267	0.313	0.965
Tier 3				
Decition Tree	0.250	0.179	0.188	0.971
Extra Trees	0.275	0.023	0.040	0.980
Random Forest	0.168	0.012	0.021	0.980
LR-OvA^{bow}	0.387	0.280	0.303	0.977
LR-OvA$^{\theta}$	0.331	0.078	0.115	0.981
LDA 100k	0.399	0.385	0.352	0.975
LDA H 100k	0.374	0.253	0.254	0.976
LDA H mutex 100k	0.375	0.252	0.262	0.977
LDA 800k	0.431	0.399	0.367	0.974

Continued on next page ...

Model	Precision	Recall	F1	Accuracy
LDA H 800k	0.396	0.265	0.264	0.976
LDA H mutex 800k	0.392	0.247	0.266	0.977
Tier 4				
Decision Tree	0.319	0.221	0.223	0.978
Extra Trees	0.659	0.125	0.203	0.986
Random Forest	0.469	0.065	0.110	0.985
LR-OvA^{bow}	0.377	0.398	0.368	0.980
LR-OvAθ	0.610	0.234	0.311	0.986
LDA 100k	0.487	0.505	0.473	0.984
LDA H 100k	0.364	0.049	0.085	0.984
LDA H mutex 100k	0.327	0.779	0.456	0.971
LDA 800k	0.503	0.503	0.486	0.985
LDA H 800k	0.326	0.019	0.034	0.985
LDA H mutex 800k	0.347	0.716	0.460	0.975

Table 27: Per category average (macro) performance metrics for labels and the different label tiers separated out. Note that the F1-score displayed is not the harmonic mean of the listed precision and recall values but the average of the individual F1-scores for each category.

F

SOFTWARE ENVIRONMENT

```
backcall                  0.1.0                    py36_0
blas                      1.0                         mkl
bleach                    2.1.3                    py36_0
blosc                     1.14.4                hdbcaa40_0
bokeh                     0.13.0                   py36_0
boto                      2.49.0                    <pip>
boto3                     1.9.0                     <pip>
botocore                  1.12.0                    <pip>
bz2file                   0.98                      <pip>
bzip2                     1.0.6                 h14c3975_5
ca-certificates           2018.03.07                   0
certifi                   2018.8.24                py36_1
chardet                   3.0.4                     <pip>
click                     6.7                      py36_0
cloudpickle               0.5.5                    py36_0
cycler                    0.10.0                   py36_0
cymem                     1.31.2                    <pip>
cytoolz                   0.9.0.1           py36h14c3975_1
cytoolz                   0.8.2                     <pip>
dask                      0.19.0                   py36_0
dask-core                 0.19.0                   py36_0
dbus                      1.13.2                h714fa37_1
decorator                 4.3.0                    py36_0
dill                      0.2.8.2                   <pip>
distributed               1.23.0                   py36_0
docutils                  0.14                      <pip>
entrypoints               0.2.3             py36h1aec115_2
expat                     2.2.5                 he0dffb1_0
fontconfig                2.13.0                h9420a91_0
freetype                  2.9.1                 h8a8886c_0
gensim                    3.5.0                     <pip>
glib                      2.56.1                h000015b_0
gmp                       6.1.2                 h6c8ec71_1
gst-plugins-base          1.14.0                hbbd80ab_1
gstreamer                 1.14.0                hb453b48_1
h5py                      2.8.0             py36h989c5e5_3
hdf5                      1.10.2                hba1933b_1
heapdict                  1.0.0                    py36_2
html5lib                  1.0.1             py36h2f9c1c0_0
icu                       58.2                  h9c2bf20_1
```

```
idna                      2.7                   <pip>
intel-openmp              2018.0.3                 0
ipykernel                 4.8.2                 py36_0
ipython                   6.4.0                 py36_0
ipython_genutils          0.2.0              py36hb52b0d5_0
ipywidgets                7.2.1                 py36_0
jedi                      0.12.0                py36_1
jinja2                    2.10               py36ha16c418_0
jmespath                  0.9.3                  <pip>
joblib                    0.12.2                py36_0
jpeg                      9b                 h024ee3a_2
jsonschema                2.6.0              py36h006f8b5_0
jupyter                   1.0.0                 py36_4
jupyter_client            5.2.3                 py36_0
jupyter_console           5.2.0              py36he59e554_1
jupyter_core              4.4.0              py36h7c827e3_0
kiwisolver                1.0.1              py36hf484d3e_0
libedit                   3.1.20170329        h6b74fdf_2
libffi                    3.2.1               hd88cf55_4
libgcc-ng                 8.2.0               hdf63c60_1
libgfortran-ng            7.2.0               hdf63c60_3
libpng                    1.6.34              hb9fc6fc_0
libsodium                 1.0.16              h1bed415_0
libstdcxx-ng              8.2.0               hdf63c60_1
libuuid                   1.0.3               h1bed415_2
libxcb                    1.13                h1bed415_1
libxml2                   2.9.8               h26e45fe_1
locket                    0.2.0                 py36_1
lzo                       2.10                h49e0be7_2
markupsafe                1.0                py36hd9260cd_1
matplotlib                2.2.3              py36hb69df0a_0
mistune                   0.8.3              py36h14c3975_1
mkl                       2018.0.3                 1
mkl_fft                   1.0.4              py36h4414c95_1
mkl_random                1.0.1              py36h4414c95_1
msgpack-python            0.5.6              py36h6bb024c_1
murmurhash                0.26.4                  <pip>
nbconvert                 5.3.1              py36hb41ffb7_0
nbformat                  4.4.0              py36h31c9010_0
ncurses                   6.1                 hf484d3e_0
notebook                  5.5.0                 py36_0
numexpr                   2.6.8              py36hd89afb7_0
numpy                     1.15.1             py36h1d66e8a_0
numpy-base                1.15.1             py36h81de0dd_0
openssl                   1.0.2p              h14c3975_0
packaging                 17.1                  py36_0
pandas                    0.23.1             py36h637b7d7_0
pandoc                    1.19.2.1            hea2e7c5_1
pandocfilters             1.4.2              py36ha6701b7_1
parso                     0.2.1                 py36_0
partd                     0.3.8                 py36_0
pathlib                   1.0.1                  <pip>
patsy                     0.5.0                 py36_0
pcre                      8.42                h439df22_0
pexpect                   4.6.0                 py36_0
pickleshare               0.7.4              py36h63277f8_0
pip                       10.0.1                py36_0
plac                      0.9.6                  <pip>
```

```
preshed                   1.0.1                    <pip>
prompt_toolkit            1.0.15          py36h17d85b1_0
psutil                    5.4.7           py36h14c3975_0
ptyprocess                0.5.2           py36h69acd42_0
pygments                  2.2.0           py36h0d3125c_0
pymongo                   3.7.0           py36h14c3975_0
pyparsing                 2.2.0                  py36_1
pyqt                      5.9.2           py36h751905a_0
pytables                  3.4.4           py36ha205bf6_0
python                    3.6.5               hc3d631a_2
python-dateutil           2.7.3                  py36_0
pytz                      2018.4                 py36_0
pyyaml                    3.13            py36h14c3975_0
pyzmq                     17.0.0          py36h14c3975_0
qt                        5.9.6               h52aff34_0
qtconsole                 4.3.1           py36h8f73b5b_0
readline                  7.0                 ha6073c6_4
regex                     2017.4.5                 <pip>
requests                  2.19.1                   <pip>
s3transfer                0.1.13                   <pip>
scikit-learn              0.19.1          py36hedc7406_0
scipy                     1.1.0           py36hfa4b5c9_1
seaborn                   0.9.0                  py36_0
selenium                  3.12.0                   <pip>
send2trash                1.5.0                  py36_0
setuptools                39.2.0                 py36_0
simplegeneric             0.8.1                  py36_2
sip                       4.19.8          py36hf484d3e_0
six                       1.11.0          py36h372c433_1
smart-open                1.6.0                    <pip>
snappy                    1.1.7               hbae5bb6_3
sortedcontainers          2.0.4                  py36_0
spacy                     1.7.5                    <pip>
sqlite                    3.23.1              he433501_0
statsmodels               0.9.0           py36h035aef0_0
tblib                     1.3.2                  py36_0
termcolor                 1.1.0                    <pip>
terminado                 0.8.1                  py36_1
testpath                  0.3.1           py36h8cadb63_0
thinc                     6.5.2                    <pip>
tk                        8.6.7               hc745277_3
toolz                     0.9.0                  py36_0
tornado                   5.0.2                  py36_0
tqdm                      4.28.1                   <pip>
traitlets                 4.3.2           py36h674d592_0
ujson                     1.35                     <pip>
urllib3                   1.23                     <pip>
wcwidth                   0.1.7           py36hdf4376a_0
webencodings              0.5.1           py36h800622e_1
wheel                     0.31.1                 py36_0
widgetsnbextension        3.2.1                  py36_0
wrapt                     1.10.11                  <pip>
xz                        5.2.4               h14c3975_4
yaml                      0.1.7               had09818_2
zeromq                    4.2.5               h439df22_0
zict                      0.1.3                  py36_0
zlib                      1.2.11              ha838bed_2
```

BIBLIOGRAPHY

Abdul-Mageed, Muhammad (2018). "Learning Subjective Language: Feature Engineered vs. Deep Models". In: (cit. on p. 19).

Aggarwal, Charu C. and ChengXiang Zhai (2012). "A Survey of Text Classification Algorithms". In: *Mining Text Data*. Ed. by Charu C. Aggarwal and ChengXiang Zhai. Boston, MA: Springer US, pp. 163–222. ISBN: 978-1-4614-3223-4. DOI: 10.1007/978-1-4614-3223-4_6. URL: https://doi.org/10.1007/978-1-4614-3223-4_6 (cit. on pp. 19, 21).

Aggarwal, Charu C. et al. (2001). "On the Surprising Behavior of Distance Metrics in High Dimensional Spaces". In: *Proceedings of the 8th International Conference on Database Theory*. ICDT '01. London, UK, UK: Springer-Verlag, pp. 420–434. ISBN: 3-540-41456-8 (cit. on p. 47).

Alaydie, Noor et al. (2012). "Exploiting Label Dependency for Hierarchical Multi-label Classification". In: *Proceedings of the 16th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part I*. PAKDD'12. Kuala Lumpur, Malaysia: Springer-Verlag, pp. 294–305. ISBN: 978-3-642-30216-9 (cit. on p. 92).

Alsumait, Loulwah et al. (2009). "Topic Significance Ranking of LDA Generative Models". In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*. ECML PKDD '09. Bled, Slovenia: Springer-Verlag, pp. 67–82. ISBN: 978-3-642-04179-2. DOI: 10.1007/978-3-642-04180-8_22. URL: http://dx.doi.org/10.1007/978-3-642-04180-8_22 (cit. on p. 43).

Aue, Anthony and Michael Gamon (2005). "Customizing Sentiment Classifiers to New Domains : a Case Study". In: *Submitted to RANLP-05, the International Conference on Recent Advances in Natural Language Processing*. Borovets, BG. URL: https://www.microsoft.com/en-us/research/publication/customizing-sentiment-classifiers-to-new-domains-a-case-study/ (cit. on p. 15).

Balahur, Alexandra et al. (2010). "The OpAL System at NTCIR 8 MOAT". In: *NTCIR* (cit. on p. 17).

Batmanghelich, Kayhan et al. (2016). "Nonparametric Spherical Topic Modeling with Word Embeddings". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 537–542. URL: http://anthology.aclweb.org/P16-2087 (cit. on pp. 37, 121).

Becker, Lee et al. (2013). "AVAYA: Sentiment Analysis on Twitter with Self-Training and Polarity Lexicon Expansion". In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 333–340. URL: http://www.aclweb.org/anthology/S13-2055 (cit. on p. 8).

Benamara, Farah, Baptiste Chardon, Yvette Yannick Mathieu, et al. (2011). "Towards Context-Based Subjectivity Analysis". In: *IJCNLP* (cit. on pp. 10, 19).

Benamara, Farah, Baptiste Chardon, Yvette Mathieu Yannick, et al. (2012). "How do Negation and Modality Impact on Opinions?" In: *ExProM@ACL* (cit. on p. 18).

Benamara, Farah et al. (2017). "Evaluative Language Beyond Bags of Words: Linguistic Insights and Computational Applications". In: *Computational Linguistics* 43, pp. 201–264 (cit. on p. 17).

Blei, David M. and John D. Lafferty (2006). "Dynamic Topic Models". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: ACM, pp. 113–120. ISBN: 1-59593-383-2. DOI: 10.1145/1143844.1143859. URL: http://doi.acm.org/10.1145/1143844.1143859 (cit. on p. 37).

Blei, David M. and Jon D. McAuliffe (2007). "Supervised Topic Models". In: *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pp. 121–128. URL: http://papers.nips.cc/paper/3328-supervised-topic-models (cit. on pp. 37–39).

Blei, David M. et al. (2003). "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3, pp. 993–1022 (cit. on pp. 22, 35–37).

Blinov, Pavel and Eugeny Kotelnikov (2014). "Blinov: Distributed Representations of Words for Aspect-Based Sentiment Analysis at SemEval 2014". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland:

Association for Computational Linguistics and Dublin City University, pp. 140–144. URL: http://www.aclweb.org/anthology/S14-2020 (cit. on p. 13).

Blitzer, John et al. (2007). "Biographies , Bollywood , Boom-boxes and Blenders : Domain Adaptation for Sentiment Classification". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. June, pp. 440–447 (cit. on pp. 3, 16, 59).

Bollegala, Danushka Tarupathi, David Weir, John Carroll, and Mitsuru Ishizuka (2010). "Cross-Domain Sentiment Classification using an Automatically Extracted Sentiment Sensitive Thesaurus". In: *EMNLP 2010 submission* (cit. on pp. 14, 54).

Bollegala, Danushka et al. (2011). "Using Multiple Sources to Construct a Sentiment Sensitive Thesaurus for Cross-Domain Sentiment Classification". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 132–141. URL: http://www.aclweb.org/anthology/P11-1014 (cit. on pp. 14, 59).

Boser, Bernhard E. et al. (1992). "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: ACM, pp. 144–152. ISBN: 0-89791-497-X. DOI: 10.1145/130385.130401. URL: http://doi.acm.org/10.1145/130385.130401 (cit. on p. 25).

Bouma, G. (2009). "Normalized (Pointwise) mutual information in collocation extraction." In: *Proceedings of the Biennial GSCL Conference, 2009, pp. 31-40.* (Cit. on p. 63).

Boutell, Matthew R. et al. (2004). "Learning multi-label scene classification". In: *Pattern Recognition* 37.9, pp. 1757–1771. DOI: 10.1016/j.patcog.2004.03.009. URL: https://doi.org/10.1016/j.patcog.2004.03.009 (cit. on pp. 46 sq., 50).

Breiman, L. (1996a). "Bagging Predictors". In: *Machine Learning* 24.2, pp. 123–140. DOI: 10.1007/BF00058655. URL: http://dx.doi.org/10.1007/BF00058655 (cit. on pp. 30–33).

— (1996b). *Out-Of-Bag Estimation*. Tech. rep. (cit. on p. 32).

— (1997). *Arcing the edge*. Tech. rep. (cit. on p. 28).

— (1999). "Pasting Small Votes for Classification in Large Databases and On-Line". In: *Machine Learning* 36.1, pp. 85–103. ISSN: 1573-0565. DOI: 10.1023/A:1007563306331. URL: https://doi.org/10.1023/A:1007563306331 (cit. on p. 32).

Breiman, L. (2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324. URL: http://dx.doi.org/10.1023/A:1010933404324 (cit. on p. 33).

Breiman, L. et al. (1984). *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis. ISBN: 9780412048418. URL: https://books.google.de/books?id=JwQx-WOmSyQC (cit. on p. 24).

Cerri, Ricardo et al. (2014). "Hierarchical multi-label classification using local neural networks". In: *Journal of Computer and System Sciences* 80.1, pp. 39–56. ISSN: 0022-0000. DOI: http://dx.doi.org/10.1016/j.jcss.2013.03.007. URL: http://www.sciencedirect.com/science/article/pii/S0022000013000718 (cit. on pp. 45, 48).

Chalothorn, Tawunrat and Jeremy Ellman (2013). "TJP: Using Twitter to Analyze the Polarity of Contexts". In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 375–379. URL: http://www.aclweb.org/anthology/S13-2061 (cit. on p. 8).

Chang, Jonathan, Jordan L. Boyd-Graber, et al. (2009). "Reading Tea Leaves: How Humans Interpret Topic Models". In: *NIPS*, pp. 288–296 (cit. on p. 43).

Chang, Ming-Wei, Lev Ratinov, et al. (2008). "Importance of Semantic Representation: Dataless Classification". In: *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*. AAAI'08. Chicago, Illinois: AAAI Press, pp. 830–835. ISBN: 978-1-57735-368-3. URL: http://dl.acm.org/citation.cfm?id=1620163.1620201 (cit. on p. 42).

Chaovalit, Pimwadee and Lina Zhou (2005). "Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches". In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 112c–112c (cit. on p. 8).

Chen, Rung-Ching and Chung-Hsun Hsieh (2006). "Web page classification based on a support vector machine using a weighted vote schema". In: *Expert Systems with Applications* 31.2, pp. 427–435. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2005.09.079. URL: http://www.sciencedirect.com/science/article/pii/S0957417405002307 (cit. on p. 41).

Chen, Xingyuan, Yunqing Xia, et al. (2015). "Dataless Text Classification with Descriptive LDA". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI'15. Austin, Texas: AAAI Press, pp. 2224–2231. ISBN: 0-262-51129-0. URL: http://dl.acm.org/citation.cfm?id=2886521.2886630 (cit. on p. 42).

Cheng, Weiwei and Eyke Hüllermeier (2009). "Combining instance-based learning and logistic regression for multilabel classification". In: *Machine Learning* 76.2, pp. 211–225. ISSN: 1573-0565. DOI: 10.1007/s10994-009-5127-5. URL: https://doi.org/10.1007/s10994-009-5127-5 (cit. on p. 49).

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018. URL: https://doi.org/10.1007/BF00994018 (cit. on p. 25).

Cover, T. and P. Hart (1967). "Nearest neighbor pattern classification". In: *IEEE Transactions on Information Theory* 13.1, pp. 21–27. ISSN: 0018-9448. DOI: 10.1109/TIT.1967.1053964 (cit. on p. 15).

Das, Rajarshi et al. (2015). "Gaussian LDA for Topic Models with Word Embeddings". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 795–804. URL: http://www.aclweb.org/anthology/P15-1077 (cit. on p. 37).

Dave, Kushal et al. (2003). "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews". In: *Proceedings of the 12th International Conference on World Wide Web*. WWW '03. Budapest, Hungary: ACM, pp. 519–528. ISBN: 1-58113-680-3. DOI: 10.1145/775152.775226. URL: http://doi.acm.org.ezproxy.sussex.ac.uk/10.1145/775152.775226 (cit. on pp. 10–12).

Deerwester, Scott C. et al. (1990). "Indexing by Latent Semantic Analysis". In: *JASIS* 41.6, pp. 391–407. DOI: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9. URL: http://dx.doi.org/10.1002/(SICI)1097-4571(199009)41:6%3C391::AID-ASI1%3E3.0.CO;2-9 (cit. on pp. 22, 34).

Dembczynski, Krzysztof and Weiwei Cheng (2010). "On Label Dependence in Multi-Label Classification". In: (cit. on p. 49).

Ding, Xiaowen et al. (2008). "A Holistic Lexicon-based Approach to Opinion Mining". In: *Proceedings of the 2008 International Conference on Web Search and Data Mining*.

WSDM '08. Palo Alto, California, USA: ACM, pp. 231–240. ISBN: 978-1-59593-927-2. DOI: 10.1145/1341531.1341561. URL: http://doi.acm.org.ezproxy.sussex.ac.uk/10.1145/1341531.1341561 (cit. on pp. 12 sq.).

Dumais, S. T. et al. (1988). "Using Latent Semantic Analysis to Improve Access to Textual Information". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '88. Washington, D.C., USA: ACM, pp. 281–285. ISBN: 0-201-14237-6. DOI: 10.1145/57167.57214. URL: http://doi.acm.org/10.1145/57167.57214 (cit. on p. 34).

Dunning, Ted (1993). "Accurate methods for the statistics of surprise and coincidence". In: *Comput. Linguist.* 19.1, pp. 61–74. ISSN: 0891-2017 (cit. on p. 11).

Efron, B. and R.J. Tibshirani (1994). *An Introduction to the Bootstrap*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis. ISBN: 9780412042317. URL: https://books.google.de/books?id=gLlpIUxRntoC (cit. on p. 32).

Forman, George (2003). "An extensive empirical study of feature selection metrics for text classification". In: *J. Mach. Learn. Res.* 3, pp. 1289–1305. ISSN: 1532-4435 (cit. on pp. 19, 21).

Freund, Yoav and Robert E Schapire (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1, pp. 119–139. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1504. URL: http://www.sciencedirect.com/science/article/pii/S002200009791504X (cit. on p. 29).

Friedman, Jerome H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of Statistics* 29.5, pp. 1189–1232. ISSN: 00905364. URL: http://www.jstor.org/stable/2699986 (cit. on pp. 29 sq.).

Galindo, Eva Lucrecia Gibaja and Sebastián Ventura (2014). "Multi-label learning: a review of the state of the art and ongoing research". In: *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery* 4, pp. 411–444 (cit. on p. 45).

García Pablos, Aitor et al. (2014). "V3: Unsupervised Generation of Domain Aspect Terms for Aspect Based Sentiment Analysis". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for

Computational Linguistics and Dublin City University, pp. 833–837. URL: http://www.aclweb.org/anthology/S14-2148 (cit. on p. 13).

Geurts, Pierre et al. (2006). "Extremely randomized trees". In: *Machine Learning* 63.1, pp. 3–42. DOI: 10.1007/s10994-006-6226-1. URL: http://dx.doi.org/10.1007/s10994-006-6226-1 (cit. on p. 33).

Griffiths, Thomas L. and Mark Steyvers (2004). "Finding scientific topics". In: *Proceedings of the National Academy of Sciences* 101.suppl 1, pp. 5228–5235. DOI: 10.1073/pnas.0307752101. eprint: http://www.pnas.org/content/101/suppl_1/5228.full.pdf. URL: http://www.pnas.org/content/101/suppl_1/5228.abstract (cit. on p. 37).

Gupta, Deepak Kumar and Asif Ekbal (2014). "IITP: Supervised Machine Learning for Aspect based Sentiment Analysis". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, pp. 319–323. URL: http://www.aclweb.org/anthology/S14-2053 (cit. on p. 13).

Guyon, Isabelle and André Elisseeff (2003). "An Introduction to Variable and Feature Selection". In: *J. Mach. Learn. Res.* 3, pp. 1157–1182. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944919.944968 (cit. on p. 21).

Guyon, Isabelle, Jason Weston, et al. (2002). "Gene Selection for Cancer Classification Using Support Vector Machines". In: *Mach. Learn.* 46.1-3, pp. 389–422. ISSN: 0885-6125. DOI: 10.1023/A:1012487302797. URL: http://dx.doi.org/10.1023/A:1012487302797 (cit. on p. 22).

Hand, David J. (2009). "Measuring classifier performance: a coherent alternative to the area under the ROC curve". In: *Mach. Learn.* 77.1, pp. 103–123. ISSN: 0885-6125. DOI: 10.1007/s10994-009-5119-5 (cit. on p. 65).

Hand, D.J. and C. Anagnostopoulos (2013). "When is the area under the receiver operating characteristic curve an appropriate measure of classifier performance?" In: *Pattern Recognition Letters* 34.5, pp. 492–495. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2012.12.004. URL: http://www.sciencedirect.com/science/article/pii/S0167865512003923 (cit. on p. 65).

Hangya, Viktor et al. (2014). "SZTE-NLP: Aspect level opinion mining exploiting syntactic cues". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics and Dublin

City University, pp. 610–614. URL: http://www.aclweb.org/anthology/S14-2107 (cit. on p. 13).

Hanley, J A and B J McNeil (1982). "The meaning and use of the area under a receiver operating characteristic (ROC) curve." In: *Radiology* 143.1. PMID: 7063747, pp. 29–36. DOI: 10.1148/radiology.143.1.7063747. eprint: https://doi.org/10.1148/radiology.143.1.7063747. URL: https://doi.org/10.1148/radiology.143.1.7063747 (cit. on p. 65).

He, Ruining and Julian McAuley (2016). "Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering". In: *CoRR* abs/1602.01585. arXiv: 1602.01585. URL: http://arxiv.org/abs/1602.01585 (cit. on p. 59).

Ho, Tin Kam (1998). "The Random Subspace Method for Constructing Decision Forests". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 20.8, pp. 832–844. DOI: 10.1109/34.709601. URL: http://dx.doi.org/10.1109/34.709601 (cit. on pp. 32 sq.).

Hoffman, Matthew D. et al. (2010). "Online Learning for Latent Dirichlet Allocation". In: *NIPS*, pp. 856–864 (cit. on p. 37).

Hofmann, Thomas (1999a). "Learning the Similarity of Documents: An Information-geometric Approach to Document Retrieval and Categorization". In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. NIPS'99. Denver, CO: MIT Press, pp. 914–920. URL: http://dl.acm.org/citation.cfm?id=3009657.3009786 (cit. on p. 35).

— (1999b). "Probabilistic Latent Semantic Indexing". In: *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '99. Berkeley, California, USA: ACM, pp. 50–57. ISBN: 1-58113-096-1. DOI: 10.1145/312624.312649. URL: http://doi.acm.org/10.1145/312624.312649 (cit. on p. 35).

Howard, Jeremy and Sebastian Ruder (2018). "Fine-tuned Language Models for Text Classification". In: *CoRR* abs/1801.06146. arXiv: 1801.06146. URL: http://arxiv.org/abs/1801.06146 (cit. on p. 20).

Hu, Minqing and Bing Liu (2004). "Mining and Summarizing Customer Reviews". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. Seattle, WA, USA: ACM, pp. 168–177. ISBN: 1-58113-888-

1. DOI: 10.1145/1014052.1014073. URL: http://doi.acm.org/10.1145/1014052.1014073 (cit. on pp. 12 sq.).

Jacobs, R. A. et al. (1991). "Adaptive Mixtures of Local Experts". In: *Neural Computation* 3.1, pp. 79–87. ISSN: 0899-7667. DOI: 10.1162/neco.1991.3.1.79 (cit. on p. 55).

Jameel, Shoaib et al. (2015). "Supervised topic models with word order structure for document classification and retrieval learning". English. In: *Information Retrieval Journal* 18.4, pp. 283–330. ISSN: 1386-4564. DOI: 10.1007/s10791-015-9254-2. URL: http://dx.doi.org/10.1007/s10791-015-9254-2 (cit. on p. 37).

Jin, Wei et al. (2009). "OpinionMiner: A Novel Machine Learning System for Web Opinion Mining and Extraction". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: ACM, pp. 1195–1204. ISBN: 978-1-60558-495-9. DOI: 10.1145/1557019.1557148. URL: http://doi.acm.org/10.1145/1557019.1557148 (cit. on p. 13).

Kakadiaris, Ioannis A. et al. (2016). "ACL 2016 The 4th BioASQ Workshop A challenge on large-scale biomedical semantic indexing and question answering". In: (cit. on pp. 45, 51).

Kamps, Jaap et al. (2004). "Using WordNet to Measure Semantic Orientations of Adjectives". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/734.pdf (cit. on p. 12).

Katakis, Ioannis et al. (2008). "Multilabel Text Classification for Automated Tag Suggestion". In: (cit. on p. 45).

Kim, Yoon (2014). "Convolutional Neural Networks for Sentence Classification". In: *EMNLP* (cit. on p. 14).

Lacoste-Julien, Simon et al. (2008). "DiscLDA: Discriminative Learning for Dimensionality Reduction and Classification". In: *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pp. 897–904. URL: http://papers.nips.cc/paper/3599-disclda-discriminative-learning-for-dimensionality-reduction-and-classification (cit. on p. 38).

Lapin, Maksim et al. (2014). "Learning using privileged information: SVM+ and weighted SVM". In: *Neural Networks* 53.Supplement C, pp. 95–108. ISSN: 0893-6080. DOI: https:

//doi.org/10.1016/j.neunet.2014.02.002. URL: http://www.sciencedirect.com/science/article/pii/S0893608014000306 (cit. on p. 27).

Lau, Jey Han et al. (2014). "Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 530–539. URL: http://www.aclweb.org/anthology/E14-1056 (cit. on p. 44).

Lauer, Fabien and Gérard Bloch (2008). "Incorporating prior knowledge in support vector machines for classification: A review". In: *Neurocomputing* 71.7. Progress in Modeling, Theory, and Application of Computational Intelligenc, pp. 1578–1594. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2007.04.010. URL: http://www.sciencedirect.com/science/article/pii/S0925231207001439 (cit. on p. 26).

Levatić, Jurica et al. (2015). "The importance of the label hierarchy in hierarchical multi-label classification". In: *Journal of Intelligent Information Systems* 45.2, pp. 247–271. ISSN: 1573-7675. DOI: 10.1007/s10844-014-0347-y. URL: http://dx.doi.org/10.1007/s10844-014-0347-y (cit. on pp. 45, 49, 51, 92).

Lewis, David D. et al. (2004). "RCV1: A New Benchmark Collection for Text Categorization Research". In: *Journal of Machine Learning Research* 5, pp. 361–397 (cit. on pp. 45, 99).

Li, Feng et al. (2017). "Granular Multi-label Feature Selection Based on Mutual Information". In: *Pattern Recogn.* 67.C, pp. 410–423. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2017.02.025. URL: https://doi.org/10.1016/j.patcog.2017.02.025 (cit. on p. 46).

Li, Wei and Andrew McCallum (2006). "Pachinko allocation: DAG-structured mixture models of topic correlations". In: *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pp. 577–584. DOI: 10.1145/1143844.1143917. URL: http://doi.acm.org/10.1145/1143844.1143917 (cit. on p. 37).

Li, Ximing et al. (2015). "Supervised topic models for multi-label classification". In: *Neurocomputing* 149, Part B, pp. 811–819. ISSN: 0925-2312. DOI: http://dx.doi.org/10.1016/j.neucom.2014.07.053. URL: http://www.sciencedirect.com/science/article/pii/S0925231214010054 (cit. on pp. 37, 48, 92).

Lin, Chenghua and Yulan He (2009). "Joint Sentiment/Topic Model for Sentiment Analysis". In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM '09. Hong Kong, China: ACM, pp. 375–384. ISBN: 978-1-60558-512-3. DOI: 10.1145/1645953.1646003. URL: http://doi.acm.org/10.1145/1645953.1646003 (cit. on p. 41).

Lin, Chun-Fu and Sheng-De Wang (2002). "Fuzzy support vector machines". In: *IEEE Transactions on Neural Networks* 13.2, pp. 464–471. ISSN: 1045-9227. DOI: 10.1109/72.991432 (cit. on p. 26).

Lin, Dekang (1998). "Automatic Retrieval and Clustering of Similar Words". In: *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*. COLING '98. Montreal, Quebec, Canada: Association for Computational Linguistics, pp. 768–774. DOI: 10.3115/980432.980696. URL: https://doi.org/10.3115/980432.980696 (cit. on p. 17).

Liu, Jingzhou et al. (2017). "Deep Learning for Extreme Multi-label Text Classification". In: *SIGIR* (cit. on p. 49).

Louppe, Gilles and Pierre Geurts (2012). "Ensembles on Random Patches". In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part I*, pp. 346–361. DOI: 10.1007/978-3-642-33460-3_28. URL: http://dx.doi.org/10.1007/978-3-642-33460-3_28 (cit. on p. 33).

Loza Mencía, Eneldo and Johannes Fürnkranz (2008). "Efficient Pairwise Multilabel Classification for Large-Scale Problems in the Legal Domain". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Walter Daelemans et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 50–65. ISBN: 978-3-540-87481-2 (cit. on pp. 45, 51).

Lyons, John (1995). *Linguistic Semantics: An Introduction*. Cambridge University Press. DOI: 10.1017/CBO9780511810213 (cit. on p. 2).

Lyra, Matti et al. (2013). "High Value Media Monitoring With Machine Learning". English. In: *KI - Künstliche Intelligenz*, pp. 1–11. ISSN: 0933-1875. DOI: 10.1007/s13218-013-0255-2 (cit. on p. 4).

Madjarov, Gjorgji et al. (2012). "An extensive experimental comparison of methods for multi-label learning". In: *Pattern Recognition* 45.9. Best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011), pp. 3084–3104. ISSN: 0031-

3203. DOI: http://dx.doi.org/10.1016/j.patcog.2012.03.004. URL: http://www.sciencedirect.com/science/article/pii/S0031320312001203 (cit. on pp. 45, 92, 94).

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press. ISBN: 0-262-13360-1 (cit. on p. 21).

Manning, Christopher D. et al. (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press. ISBN: 0521865719, 9780521865715 (cit. on pp. 19, 21).

Mason, Llew et al. (1999). "Boosting Algorithms As Gradient Descent". In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. NIPS'99. Denver, CO: MIT Press, pp. 512–518. URL: http://dl.acm.org/citation.cfm?id=3009657.3009730 (cit. on pp. 29 sq.).

Matthews, B.W. (1975). "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA) - Protein Structure* 405.2, pp. 442–451. ISSN: 0005-2795. DOI: https://doi.org/10.1016/0005-2795(75)90109-9. URL: http://www.sciencedirect.com/science/article/pii/0005279575901099 (cit. on p. 64).

McCallum, Andrew Kachites (1999). "Multi-label text classification with a mixture model trained by EM". In: *AAAI 99 Workshop on Text Learning* (cit. on p. 47).

McCullagh, P. and John A. Nelder (1989). *Generalized linear models*. English. London; New York: Chapman and Hall (cit. on p. 38).

Mei, Qiaozhu et al. (2007). "Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs". In: *Proceedings of the 16th International Conference on World Wide Web*. WWW '07. Banff, Alberta, Canada: ACM, pp. 171–180. ISBN: 978-1-59593-654-7. DOI: 10.1145/1242572.1242596. URL: http://doi.acm.org/10.1145/1242572.1242596 (cit. on p. 41).

Mikolov, Tomas, Kai Chen, et al. (2013). "Efficient Estimation of Word Representations in Vector Space". In: *CoRR* abs/1301.3781 (cit. on p. 14).

Mikolov, Tomas, Ilya Sutskever, et al. (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *NIPS*, pp. 3111–3119 (cit. on pp. 14, 117).

Miller, George A. et al. (1990). "Introduction to WordNet: An On-line Lexical Database*". In: *International Journal of Lexicography* 3.4, pp. 235–244. DOI: 10.1093/ijl/3.4.235. eprint: /oup/backfile/content_public/journal/ijl/3/4/10.1093_ijl_3.4.235/1/235.pdf. URL: http://dx.doi.org/10.1093/ijl/3.4.235 (cit. on p. 12).

Mimno, David and David Blei (2011). "Bayesian Checking for Topic Models". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, pp. 227–237. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145459 (cit. on p. 44).

Morinaga, Satoshi et al. (2002). "Mining product reputations on the Web". In: *KDD* (cit. on p. 10).

Mork, James G. et al. (2017). "12 years on – Is the NLM medical text indexer still useful and relevant?" In: *J. Biomedical Semantics* (cit. on p. 51).

Mork, J.G. et al. (2013). "The NLM medical text indexer system for indexing biomedical literature". In: *CEUR Workshop Proceedings* 1094 (cit. on p. 51).

Morstatter, Fred and Huan Liu (2016). "A Novel Measure for Coherence in Statistical Topic Models". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 543–548. URL: http://anthology.aclweb.org/P16-2088 (cit. on p. 44).

Negi, Sapna and Paul Buitelaar (2014). "INSIGHT Galway: Syntactic and Lexical Features for Aspect Based Sentiment Analysis". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, pp. 346–350. URL: http://www.aclweb.org/anthology/S14-2058 (cit. on p. 13).

Nigam, Kamal et al. (2000). "Text Classification from Labeled and Unlabeled Documents using EM". In: *Machine Learning* 39.2, pp. 103–134. ISSN: 1573-0565. DOI: 10.1023/A:1007692713085. URL: https://doi.org/10.1023/A:1007692713085 (cit. on p. 15).

Pan, Sinno Jialin et al. (2010). "Cross-Domain Sentiment Classification via Spectral Feature Alignment". In: *WWW 2010* (cit. on p. 59).

Pang, Bo and Lillian Lee (2004). "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *Proceedings of the 42Nd*

Annual Meeting on Association for Computational Linguistics. ACL '04. Barcelona, Spain: Association for Computational Linguistics. DOI: 10.3115/1218955.1218990. URL: https://doi.org/10.3115/1218955.1218990 (cit. on p. 8).

Pang, Bo and Lillian Lee (2005). "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales". In: *ACL* (cit. on p. 13).

Pang, Bo et al. (2002). "Thumbs Up?: Sentiment Classification Using Machine Learning Techniques". In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 79–86. DOI: 10.3115/1118693.1118704. URL: https://doi.org/10.3115/1118693.1118704 (cit. on pp. 8 sq., 12, 18).

Papagiannopoulou, Eirini et al. (2016). "Large-Scale Semantic Indexing and Question Answering in Biomedicine". In: (cit. on p. 51).

Pennington, Jeffrey et al. (2014). "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162 (cit. on p. 14).

Polanyi, Livia and Annie Zaenen (2006). "Contextual Valence Shifters". In: *Computing Attitude and Affect in Text* (cit. on pp. 10, 17 sq.).

Ponomareva, Natalia and Mike Thelwall (2013). "Semi-supervised vs. Cross-domain Graphs for Sentiment Analysis". In: *Recent Advances in Natural Language Processing, RANLP 2013, 9-11 September, 2013, Hissar, Bulgaria*, pp. 571–578. URL: http://aclweb.org/anthology/R/R13/R13-1075.pdf (cit. on pp. 15 sq.).

Pontiki, Maria et al. (2014). "SemEval-2014 Task 4: Aspect Based Sentiment Analysis". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, pp. 27–35. URL: http://www.aclweb.org/anthology/S14-2004 (cit. on p. 13).

Popescul, Alexandrin et al. (2001). "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments". In: *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. UAI '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 437–444. ISBN: 1-55860-800-1. URL: http://dl.acm.org/citation.cfm?id=647235.720088 (cit. on p. 35).

Poursepanj, Hamid et al. (2013). "uOttawa: System description for SemEval 2013 Task 2 Sentiment Analysis in Twitter". In: *Second Joint Conference on Lexical and Computa-*

*tional Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 380–383. URL: http://www.aclweb.org/anthology/S13-2062 (cit. on p. 8).

Questier, F. et al. (2005). "The use of CART and multivariate regression trees for supervised and unsupervised feature selection". In: *Chemometrics and Intelligent Laboratory Systems* 76.1, pp. 45–54. ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2004.09.003. URL: http://www.sciencedirect.com/science/article/pii/S0169743904002187 (cit. on p. 24).

Quinlan, J. R. (1986). "Induction of decision trees". In: *Machine Learning* 1.1, pp. 81–106. ISSN: 1573-0565. DOI: 10.1007/BF00116251. URL: https://doi.org/10.1007/BF00116251 (cit. on p. 23).

Quinlan, J. Ross (1993). *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1558602402 (cit. on p. 23).

Ramage, Daniel et al. (2009). "Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*. EMNLP '09. Singapore: Association for Computational Linguistics, pp. 248–256. ISBN: 978-1-932432-59-6. URL: http://dl.acm.org/citation.cfm?id=1699510.1699543 (cit. on p. 40).

Read, Jesse et al. (2011). "Classifier chains for multi-label classification". In: *Machine Learning* 85.3, p. 333. ISSN: 1573-0565. DOI: 10.1007/s10994-011-5256-5. URL: https://doi.org/10.1007/s10994-011-5256-5 (cit. on p. 50).

Read, Jonathon (2005). "Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification". In: *Proceedings of the ACL Student Research Workshop*. ACLstudent '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 43–48. URL: http://dl.acm.org/citation.cfm?id=1628960.1628969 (cit. on pp. 15 sq.).

Recasens, Marta et al. (2013). "Linguistic Models for Analyzing and Detecting Biased Language". In: *ACL* (cit. on p. 18).

Riloff, Ellen and Janyce Wiebe (2003). "Learning Extraction Patterns for Subjective Expressions". In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. EMNLP '03. Stroudsburg, PA, USA: Association for Computational

Linguistics, pp. 105–112. DOI: 10.3115/1119355.1119369. URL: https://doi.org/10.3115/1119355.1119369 (cit. on p. 17).

Rissanen, J. J. (1996). "Fisher information and stochastic complexity". In: *IEEE Transactions on Information Theory* 42.1, pp. 40–47. ISSN: 0018-9448. DOI: 10.1109/18.481776 (cit. on p. 10).

Röder, Michael et al. (2015). "Exploring the Space of Topic Coherence Measures". In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. WSDM '15. Shanghai, China: ACM, pp. 399–408. ISBN: 978-1-4503-3317-7. DOI: 10.1145/2684822.2685324. URL: http://doi.acm.org/10.1145/2684822.2685324 (cit. on p. 44).

Rosen-Zvi, Michal et al. (2004). "The Author-topic Model for Authors and Documents". In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. UAI '04. Banff, Canada: AUAI Press, pp. 487–494. ISBN: 0-9749039-0-6. URL: http://dl.acm.org/citation.cfm?id=1036843.1036902 (cit. on pp. 38 sq., 48).

Rubin, Timothy N. et al. (2012). "Statistical topic models for multi-label document classification". In: *Machine Learning* 88.1-2, pp. 157–208. DOI: 10.1007/s10994-011-5272-5. URL: http://dx.doi.org/10.1007/s10994-011-5272-5 (cit. on pp. 48, 92).

Al-Salemi, Bassam et al. (2015). "LDA-AdaBoost.MH: Accelerated AdaBoost.MH based on latent Dirichlet allocation for text categorization". In: *J. Information Science* 41, pp. 27–40 (cit. on p. 47).

Schapire, Robert E. and Yoram Singer (2000). "BoosTexter: A Boosting-based System for Text Categorization". In: *Machine Learning* 39.2, pp. 135–168. ISSN: 1573-0565. DOI: 10.1023/A:1007649029923. URL: https://doi.org/10.1023/A:1007649029923 (cit. on pp. 45, 47).

Segura-Bedmar, Isabel and Adrian Carruana (2016). "LABDA at the 2016 BioASQ challenge task 4 a : Semantic Indexing by using ElasticSearch". In: (cit. on p. 51).

Seki, Yohei, David Kirk Evans, Lun-Wei Ku, Hsin-Hsi Chen, et al. (2007). "Overview of Opinion Analysis Pilot Task at NTCIR-6". In: *NTCIR* (cit. on p. 16).

Seki, Yohei, David Kirk Evans, Lun-Wei Ku, Le Sun, et al. (2008). "Overview of Multilingual Opinion Analysis Task at NTCIR-7". In: *NTCIR* (cit. on p. 16).

Seki, Yohei, Lun-Wei Ku, et al. (2010). "Overview of Multilingual Opinion Analysis Task at NTCIR-8: A Step Toward Cross Lingual Opinion Analysis". In: *NTCIR* (cit. on p. 16).

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press. ISBN: 1107057132, 9781107057135 (cit. on pp. 19, 24).

Shalev-Shwartz, Shai, Yoram Singer, et al. (2011). "Pegasos: Primal Estimated Sub-gradient Solver for SVM". In: *Math. Program.* 127.1, pp. 3–30. ISSN: 0025-5610. DOI: 10.1007/s10107-010-0420-4. URL: http://dx.doi.org/10.1007/s10107-010-0420-4 (cit. on p. 26).

Shams, Mohammadreza and Ahmad Baraani-Dastjerdi (2017). "Enriched LDA (ELDA): Combination of latent Dirichlet allocation with word co-occurrence analysis for aspect extraction". In: *Expert Systems with Applications* 80.Supplement C, pp. 136–146. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2017.02.038. URL: http://www.sciencedirect.com/science/article/pii/S095741741730129X (cit. on p. 43).

Socher, Richard et al. (2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *EMNLP* (cit. on pp. 10, 13).

Soleimani, Hossein and David J. Miller (2016). "Semi-supervised Multi-Label Topic Models for Document Classification and Sentence Labeling". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM '16. Indianapolis, Indiana, USA: ACM, pp. 105–114. ISBN: 978-1-4503-4073-1. DOI: 10.1145/2983323.2983752. URL: http://doi.acm.org.ezproxy.sussex.ac.uk/10.1145/2983323.2983752 (cit. on p. 40).

— (2017). "Semisupervised, Multilabel, Multi-Instance Learning for Structured Data". In: *Neural Computation* 29.4. PMID: 28095193, pp. 1053–1102. DOI: 10.1162/NECO_a_00939. eprint: https://doi.org/10.1162/NECO_a_00939. URL: https://doi.org/10.1162/NECO_a_00939 (cit. on p. 41).

Song, Yangqiu and Dan Roth (2014). "On Dataless Hierarchical Text Classification". In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI'14. Québec City, Québec, Canada: AAAI Press, pp. 1579–1585. URL: http://dl.acm.org/citation.cfm?id=2892753.2892772 (cit. on p. 42).

Srinivas, M et al. (2009). "Combining the Classifiers and LSI Method for Efficient and Accurate Text Classification". In: *Journal of Information Technology and Knowledge Management* 2.2, pp. 263–267 (cit. on p. 104).

Stone, Benjamin et al. (2010). "Comparing Methods for Single Paragraph Similarity Analysis". In: *Topics in Cognitive Science* 3.1, pp. 92–122. DOI: 10.1111/j.1756-8765.2010.01108.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1756-8765.2010.01108.x (cit. on p. 66).

Sugumaran, V. et al. (2007). "Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing". In: *Mechanical Systems and Signal Processing* 21.2, pp. 930–942. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2006.05.004. URL: http://www.sciencedirect.com/science/article/pii/S0888327006001142 (cit. on p. 24).

Teh, Yee Whye et al. (2006). "Hierarchical Dirichlet Processes". English. In: *Journal of the American Statistical Association* 101.476, pp. 1566–1581. ISSN: 01621459. URL: http://www.jstor.org/stable/27639773 (cit. on p. 37).

Tsoumakas, Grigorios and Ioannis Katakis (2007). "Multi-Label Classification: An Overview." In: *International Journal of Data Warehousing and Mining* 3.3, pp. 1–13. DOI: 10.4018/jdwm.2007070101 (cit. on pp. 50, 93).

Tsoumakas, Grigorios and Ioannis Vlahavas (2007). "Random k-Labelsets: An Ensemble Method for Multilabel Classification". In: *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings*. Ed. by Joost N. Kok et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 406–417. ISBN: 978-3-540-74958-5. DOI: 10.1007/978-3-540-74958-5_38. URL: http://dx.doi.org/10.1007/978-3-540-74958-5_38 (cit. on p. 47).

Tsoumakas, Grigorios et al. (2008). "Effective and Efficient Multilabel Classification in Domains with Large Number of Labels". In: *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*. Antwerp, Belgium (cit. on pp. 45 sq.).

— (2010). "Mining Multi-label Data". In: *Data Mining and Knowledge Discovery Handbook*. Ed. by Oded Maimon and Lior Rokach. Boston, MA: Springer US, pp. 667–685. ISBN: 978-0-387-09823-4. DOI: 10.1007/978-0-387-09823-4_34. URL: https://doi.org/10.1007/978-0-387-09823-4_34 (cit. on p. 105).

Turney, Peter D. (2002). "Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 417–424. DOI: 10.3115/1073083.1073153. URL: https://doi.org/10.3115/1073083.1073153 (cit. on pp. 8 sq., 12, 54).

Van Canneyt, Steven et al. (2015). "Topic-Dependent Sentiment Classification on Twitter". In: *Advances in Information Retrieval: 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015. Proceedings*. Ed. by Allan Hanbury et al. Cham: Springer International Publishing, pp. 441–446. ISBN: 978-3-319-16354-3. DOI: 10.1007/978-3-319-16354-3_48. URL: https://doi.org/10.1007/978-3-319-16354-3_48 (cit. on pp. 42, 116 sq.).

Vapnik, Vladimir and Akshay Vashist (2009). "A new learning paradigm: Learning using privileged information". In: *Neural Networks* 22.5. Advances in Neural Networks Research: IJCNN2009, pp. 544–557. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2009.06.042. URL: http://www.sciencedirect.com/science/article/pii/S0893608009001130 (cit. on p. 27).

W. Olver, Frank et al. (2010). "NIST Handbook of Mathematical Functions". In: (cit. on p. 36).

Wallach, Hanna M. (2006). "Topic Modeling: Beyond Bag-of-words". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: ACM, pp. 977–984. ISBN: 1-59593-383-2. DOI: 10.1145/1143844.1143967. URL: http://doi.acm.org/10.1145/1143844.1143967 (cit. on p. 37).

Wallach et al. (2009). "Rethinking LDA: Why Priors Matter". In: *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.* Pp. 1973–1981. URL: http://papers.nips.cc/paper/3854-rethinking-lda-why-priors-matter (cit. on p. 37).

Wang, Shuai et al. (2016). "Mining Aspect-Specific Opinion Using a Holistic Lifelong Topic Model". In: *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, pp. 167–176. ISBN: 978-1-4503-4143-1. DOI: 10.1145/

2872427.2883086. URL: https://doi-org.ezproxy.sussex.ac.uk/10.1145/2872427.2883086 (cit. on p. 43).

Wang, Sida and Christopher D. Manning (2012). "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*. ACL '12. Jeju Island, Korea: Association for Computational Linguistics, pp. 90–94. URL: http://dl.acm.org/citation.cfm?id=2390665.2390688 (cit. on p. 26).

Wang, Xuerui and Andrew McCallum (2006). "Topics over time: a non-Markov continuous-time model of topical trends". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '06. Philadelphia, PA, USA: ACM, pp. 424–433. ISBN: 1-59593-339-5. DOI: 10.1145/1150402.1150450 (cit. on p. 37).

Wiebe, Janyce et al. (2004). "Learning Subjective Language". In: *Computational Linguistics* 30, pp. 277–308 (cit. on p. 17).

Wilson, Theresa et al. (2005). "Recognizing Contextual Polarity in Phrase-level Sentiment Analysis". In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. HLT '05. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 347–354. DOI: 10.3115/1220575.1220619. URL: https://doi.org/10.3115/1220575.1220619 (cit. on p. 17).

Wu, Fangzhao et al. (2017). "Domain-specific sentiment classification via fusing sentiment knowledge from multiple sources". In: *Information Fusion* 35, pp. 26–37. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2016.09.001. URL: http://www.sciencedirect.com/science/article/pii/S1566253516300653 (cit. on pp. 16, 59).

Wu, Qiong, Songbo Tan, et al. (2009). "SentiRank: Cross-Domain Graph Ranking for Sentiment Classification". In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*. WI-IAT '09. Washington, DC, USA: IEEE Computer Society, pp. 309–314. ISBN: 978-0-7695-3801-3. DOI: 10.1109/WI-IAT.2009.55. URL: https://doi.org/10.1109/WI-IAT.2009.55 (cit. on pp. 15 sq.).

Xiang, Bing and Liang Zhou (2014). "Improving Twitter Sentiment Analysis with Topic-Based Mixture Modeling and Semi-Supervised Training". In: *Proceedings of the 52nd*

Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers, pp. 434–439. URL: http://aclweb.org/anthology/P/P14/P14-2071.pdf (cit. on pp. 5, 42, 55, 57–59, 83 sq., 89, 116 sq.).

Ye, Zhe et al. (2018). "Encoding Sentiment Information into Word Vectors for Sentiment Analysis". In: *COLING* (cit. on p. 14).

Yi, Jeonghee et al. (2003). "Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques". In: *ICDM* (cit. on p. 11).

Zavorin, Ilya (2016). "Using Learning-To-Rank to Enhance NLM Medical Text Indexer Results". In: (cit. on p. 51).

Zhang, Min-Ling and Zhi-Hua Zhou (2007). "ML-KNN: A lazy learning approach to multi-label learning". In: *Pattern Recognition* 40.7, pp. 2038–2048. ISSN: 0031-3203. DOI: http://dx.doi.org/10.1016/j.patcog.2006.12.019. URL: http://www.sciencedirect.com/science/article/pii/S0031320307000027 (cit. on pp. 47, 93).

Zhang, Wenjie, Liwei Wang, et al. (2018). "Deep Extreme Multi-label Learning". In: *ICMR* (cit. on p. 50).

Zhang, Yuhong, Xuegang Hu, et al. (2015). "Cross-domain sentiment classification-feature divergence, polarity divergence or both?" In: *Pattern Recognition Letters* 65.Supplement C, pp. 44–50. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2015.07.006. URL: http://www.sciencedirect.com/science/article/pii/S0167865515002093 (cit. on pp. 14, 54).

Zhong, Shaobo and Dongsheng Zou (2011). "Web Page Classification using an ensemble of support vector machine classifiers". In: *JNW* 6.11, pp. 1625–1630. DOI: 10.4304/jnw.6.11.1625-1630. URL: https://doi.org/10.4304/jnw.6.11.1625-1630 (cit. on p. 41).

Zhou, Guangyou et al. (2015). "Cross-domain sentiment classification via topical correspondence transfer". In: *Neurocomputing* 159.Supplement C, pp. 298–305. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2014.12.006. URL: http://www.sciencedirect.com/science/article/pii/S0925231214016701 (cit. on p. 14).

Zhu, Jun, Ning Chen, et al. (2013). "Gibbs Max-margin Topic Models with Fast Sampling Algorithm". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org,

pp. I-124–I-132. URL: http://dl.acm.org/citation.cfm?id=3042817.3042833 (cit. on p. 39).

Zhu, Jun et al. (2012). "MedLDA: maximum margin supervised topic models". In: *Journal of Machine Learning Research* 13, pp. 2237–2278. URL: http://dl.acm.org/citation.cfm?id=2503315 (cit. on pp. 37, 39).

Zhu, Xiaojin and Zoubin Ghahramani (2002). *Learning from Labeled and Unlabeled Data with Label Propagation*. Tech. rep. (cit. on p. 16).